

**Oracle® Retail Fiscal Management/RMS Brazil  
Localization**

Implementation Guide

Release 14.2

**F29404-01**

November 2020

F29404-01

Copyright © 2020, Oracle and/or its affiliates. All rights reserved.

Primary Author:

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

#### **Value-Added Reseller (VAR) Language**

##### **Oracle Retail VAR Applications**

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

- (i) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.
- (ii) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.
- (iii) the software component known as **Access Via**<sup>™</sup> licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.
- (iv) the software component known as **Adobe Flex**<sup>™</sup> licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all

reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.



---

---

# Contents

<b>Send Us Your Comments</b> .....	xv
<b>Preface</b> .....	xvii
Audience .....	xvii
Documentation Accessibility .....	xvii
Related Documents .....	xvii
Customer Support .....	xviii
Review Patch Documentation .....	xviii
Improved Process for Oracle Retail Documentation Corrections .....	xviii
Oracle Retail Documentation on the Oracle Technology Network .....	xviii
Conventions .....	xix
<b>1 Oracle Retail Fiscal Management Overview</b>	
<b>What is Oracle Retail Fiscal Management?</b> .....	1-1
System Setup .....	1-1
System Options .....	1-2
Fiscal Number .....	1-2
Tolerances .....	1-2
Requisition Types .....	1-2
Fiscal Configuration .....	1-2
Fiscal Document Type .....	1-2
Fiscal Utilization Code .....	1-2
Fiscal Utilization - Document Type .....	1-3
Fiscal Utilization - Reason Code .....	1-3
ORFM Functional Assumptions .....	1-3
<b>What is Nota Fiscal (NF)?</b> .....	1-4
<b>What is Nota Fiscal Eletrônica (NF-e)?</b> .....	1-4
<b>What is Retail Tax Integration Layer?</b> .....	1-4
<b>What is TaxWeb Tax Engine?</b> .....	1-5
<b>What is Sistema Público de Escrituração Digital (SPED)?</b> .....	1-5
<b>2 Understanding ORFM Implementation Checklists</b>	
Prerequisites .....	2-1
Implementation Checklist for Legacy Systems .....	2-1
Implementation Checklist for Standard Systems .....	2-3

### 3 Setting up RMS for ORFM and Brazil Localization

<b>Decoupling RMS and ORFM Batch</b> .....	3-1
Dynamic Function Calls .....	3-1
PL/SQL Package Calls .....	3-2
PL/SQL Type Inheritance .....	3-2
Localization Layer.....	3-2
Accomplishing Decoupling.....	3-3
Decoupling Batch Programs.....	3-3
<b>Set up Item Fiscal Attributes</b> .....	3-4
<b>Set up Fiscal Attributes for other Entities</b> .....	3-4
<b>Set up Utilization</b> .....	3-4
<b>Set up Document Types</b> .....	3-5
<b>Set up System Options</b> .....	3-5
<b>Set up Reason Codes</b> .....	3-5
<b>Set up Partner</b> .....	3-5
<b>Set Primary Country</b> .....	3-5
<b>Set VAT Indicator</b> .....	3-5
<b>Set Default Tax Type</b> .....	3-6
<b>Verify Localized Indicator</b> .....	3-6
<b>Set up Tax Codes</b> .....	3-7
<b>Set up Flex Field Attribute</b> .....	3-7
<b>Set up Fiscal Document Chunk Size</b> .....	3-8
<b>Set up FM Tran Code</b> .....	3-8
<b>Set up Set Of Books</b> .....	3-11
<b>Set up Chart of Account</b> .....	3-11
<b>Set up GL Options</b> .....	3-11
<b>Dynamic Segments Set up for Chart of Accounts</b> .....	3-11
<b>Account Set up for Chart of Account</b> .....	3-11
<b>Set up GL Cross Reference</b> .....	3-11
<b>Including Balance to RTV Balance Control</b> .....	3-11
<b>Transfer Ownership</b> .....	3-12
PO Receiving at a Warehouse .....	3-12
PO Receiving at a Warehouse .....	3-13

### 4 Integrating ORFM

<b>Purchase Order Receiving</b> .....	4-2
PO Receiving at a Warehouse .....	4-3
PO Receiving at a Store .....	4-5
<b>Transfer Shipment and Receiving</b> .....	4-6
Transfers .....	4-6
Transfer Creation .....	4-6
Transfer Shipment at Source .....	4-7
Transfer Receiving at Destination .....	4-8
<b>Return to Vendor (RTV) Shipments</b> .....	4-9
RTV Shipped from Warehouse .....	4-9
RTV Shipped from Store .....	4-10
Balance Control for RTV .....	4-11

<b>Retail Merchandise Authorization</b> .....	4-12
<b>Inventory Adjustment</b> .....	4-12
<b>Finishers/Repairing Flow (Two-Legged Transfer)</b> .....	4-13
Two-Legged Transfer Shipping from RWMS .....	4-13
Two-Legged Transfer Receiving at RWMS .....	4-15
Two-Legged Transfer Shipping from SIM .....	4-16
Two-Legged Transfer Receiving at SIM .....	4-17
<b>RIB-Based Integration</b> .....	4-18
<b>HTTP-Based Integration for TaxWeb</b> .....	4-19
<b>Nota Fiscal – Receiving and Issuing</b> .....	4-19
Nota Fiscal Receiving .....	4-19
Inbound Process Flow .....	4-19
Nota Fiscal Issuing .....	4-20
Outbound Process Flow .....	4-20
Triangulation .....	4-21
EDI NF .....	4-21
<b>ORFM and RWMS</b> .....	4-21
PO Receiving .....	4-21
Transfer Receiving .....	4-21
<b>ORFM and SIM</b> .....	4-22
<b>ORFM Integration with Financial Application</b> .....	4-22
Supplier Set up and Integration .....	4-23
ORFM AP Integration Stage Head and Detail .....	4-25

## 5 Publication and Subscription API Designs

<b>Secondary ASNOut Publication API</b> .....	5-1
Functionality Checklist .....	5-1
Form Impact .....	5-1
Business Object Records .....	5-1
Package Impact .....	5-2
Business Object ID .....	5-2
Package Specification – Global Variables .....	5-2
Function Level Description .....	5-2
Trigger Impact .....	5-4
Trigger Name .....	5-4
Trigger File Name .....	5-4
Table .....	5-4
Message XSD .....	5-4
Table Impact .....	5-4
Design Assumptions .....	5-4
<b>POSchedule Publication API</b> .....	5-5
Functionality Checklist .....	5-5
Form Impact .....	5-5
Business Object Records .....	5-5
Package Impact .....	5-5
Business Object ID .....	5-5
Package Specification - Global Variables .....	5-5

Function Level Description .....	5-5
Trigger Impact .....	5-7
Trigger Name.....	5-7
Trigger File Name .....	5-7
Table .....	5-7
Message XSD.....	5-7
Table Impact.....	5-8
Design Assumptions.....	5-8
<b>Shipment Information Subscription API.....</b>	<b>5-8</b>
Business Overview.....	5-8
Assumptions .....	5-8
Functionality Checklist.....	5-9
Form Impact.....	5-9
Business Object Records.....	5-9
Package Impact.....	5-9
Package Specification - Global Variables.....	5-9
Function/Procedure Level Description.....	5-9
Trigger Impact .....	5-10
Trigger Name.....	5-10
Trigger File Name .....	5-10
Table .....	5-10
Message XSD.....	5-10
Table Impact.....	5-10
Design Assumptions.....	5-10

## 6 Archiving and Purging Data

## 7 Integrating with TaxWeb Tax Rules– Retail Tax Integration Layer

<b>Integration Overview .....</b>	<b>7-2</b>
<b>Retail Tax Data Model.....</b>	<b>7-2</b>
Object Structure Overview .....	7-3
Retail Tax Data Model to TaxWeb Tax Rules Data Model Object Mapping .....	7-4
<b>RTIL Architecture.....</b>	<b>7-4</b>
High Level Integration View .....	7-4
RTIL Integration Architecture .....	7-5
Components Deployment View.....	7-6
Client Components .....	7-7
<b>RTIL Configuration .....</b>	<b>7-7</b>
Logging/Audit .....	7-7
LoggerAspect.....	7-8
ObjectLoggerAspect .....	7-8
TimerLoggerAspect.....	7-8
User-friendly Exception Messages .....	7-9
<b>Input Parameters Data Mapping and Expected Output Results.....</b>	<b>7-9</b>
<b>Concurrent Processing in RTIL .....</b>	<b>7-10</b>
<b>Tax. Calculation Scenarios.....</b>	<b>7-12</b>
Scenario 1 - Inbound NF through User Entry .....	7-12

Scenario 2 - Inbound PO NFs through EDI.....	7-12
Scenario 3 - PO Receiving .....	7-12
Scenario 4 - Outbound NF Issuing .....	7-12
Scenario 4 - PO Tax Calculation .....	7-13
Scenario 5 - Item Creation .....	7-13

## 8 Working with RTIL

<b>Operational Considerations</b> .....	8-1
RTIL Log File Monitoring .....	8-1
Log File Archive and Purge.....	8-2
Time-out Configurations.....	8-2
Inventory Adjustment.....	8-2
URLInvoker Time-out .....	8-3
RIB Transaction Time-out .....	8-3
RTIL WorkManager Configurations .....	8-4
Maximum Threads Constraint.....	8-4
Capacity Constraint .....	8-4
General Troubleshooting for RTIL/TaxWeb Tax Rules.....	8-4
RMS/ORFM is configured with an incorrect RTIL URL (Service Access Point) .....	8-5
RTIL Service is not responding.....	8-5
RTIL URL Configuration is correct but insufficient privileges in RMS database for RMS/ORFM to access RTIL .....	8-5
Incomplete Deployment of TaxWeb Tax Rules libraries.....	8-6
How do I know if RTIL is installed correctly? .....	8-6
Recovery in case of RTIL Compensation API failure.....	8-7
RTIL and TaxWeb Tax Rules Outage Sequence/Start and Stop.....	8-8
Probable Systemic Failure Scenarios in Tax Flows and Recovery .....	8-8
TaxWeb Failures and Associated RIB Impacts .....	8-9
Deploying RTIL .....	8-9

## 9 Understanding NF-e and SPED

<b>NF-e Solution</b> .....	9-1
NF-e Options.....	9-2
<b>NF-e Events</b> .....	9-2
<b>Webservices</b> .....	9-3
<b>Sistema Público de Escrituração Digital (SPED) Solution</b> .....	9-4
Block K - Stock Position.....	9-6
Main Flow .....	9-6
Performance Optimization for SPED .....	9-6

## 10 Maintaining Tax Related Data - TaxWeb

<b>Installing Rules from TaxWeb Tax Rules</b> .....	10-1
Item Cost for Non-Brazil Supplier .....	10-1
Program Flow .....	10-2
<b>Maintaining External Tax Information (L10N_BR_T_EXTAX_MAINT)</b> .....	10-3

## 11 Maintaining Tax Related Data - TaxWeb

<b>TaxWeb Tax Rules Integration and Maintaining Batches</b> .....	11-2
fmtaxchg (Tax Changing Process) .....	11-2
Design Overview .....	11-2
Scheduling Constraints .....	11-3
Restart/Recovery .....	11-3
Key Tables Affected .....	11-3
fmtaxupld (Tax Change Upload) .....	11-3
Design Overview .....	11-4
Scheduling Constraints .....	11-4
Restart/Recovery .....	11-4
Key Tables Affected .....	11-4
Input File .....	11-4
l10n_br_exec_tax_recalc .....	11-6
Design Overview .....	11-6
Scheduling Constraints .....	11-7
Restart/Recovery .....	11-7
Key Tables Affected .....	11-7
l10n_br_fiscal_item_reclass_cost .....	11-7
Design Overview .....	11-8
Scheduling Constraints .....	11-8
Restart/Recovery .....	11-8
Key Tables Affected .....	11-8
l10n_br_fiscal_loc_reclass_cost .....	11-9
Design Overview .....	11-9
Scheduling Constraints .....	11-9
Restart/Recovery .....	11-10
Key Tables Affected .....	11-10
l10n_br_fiscal_loc_reclass_retail .....	11-10
Design Overview .....	11-10
Scheduling Constraints .....	11-11
Restart/Recovery .....	11-11
Key Tables Affected .....	11-11
l10n_br_fiscal_reclass_item_finish_retail .....	11-11
Design Overview .....	11-11
Scheduling Constraints .....	11-12
Restart/Recovery .....	11-12
Key Tables Affected .....	11-12
l10n_br_fiscal_reclass_item_process_retail .....	11-13
Design Overview .....	11-13
Scheduling Constraints .....	11-13
Restart/Recovery .....	11-13
Key Tables Affected .....	11-14
l10n_br_fiscal_reclass_item_setup_retail .....	11-14
Design Overview .....	11-14
Scheduling Constraints .....	11-14
Restart/Recovery .....	11-14

Key Tables Affected.....	11-15
l10n_br_refresh_extax_finish_retail.....	11-16
Design Overview .....	11-16
Scheduling Constraints .....	11-16
Restart/Recovery .....	11-16
Key Tables Affected.....	11-16
l10n_br_refresh_extax_future_cost.....	11-17
Design Overview .....	11-17
Scheduling Constraints .....	11-17
Restart/Recovery .....	11-18
Key Tables Affected.....	11-18
l10n_br_refresh_extax_process_retail .....	11-20
Design Overview .....	11-20
Scheduling Constraints .....	11-20
Restart/Recovery .....	11-20
Key Tables Affected.....	11-21
l10n_br_refresh_extax_setup_retail.....	11-21
Design Overview .....	11-21
Scheduling Constraints .....	11-21
Restart/Recovery .....	11-22
Key Tables Affected.....	11-22
l10nbr_taxweb_fisdnld (Taxweb Fiscal Download).....	11-24
Design Overview .....	11-24
Scheduling Constraints .....	11-25
Restart/Recovery .....	11-25
Key Tables Affected.....	11-25
<b>Financial Posting Batches</b> .....	11-26
fmtrandata (FM Transaction Data).....	11-26
Design Overview .....	11-26
Scheduling Constraints .....	11-26
Restart/Recovery .....	11-26
Key Tables Affected.....	11-26
fmfinpost (FM Financial Posting) .....	11-26
Design Overview .....	11-27
Scheduling Constraints .....	11-27
Restart/Recovery .....	11-27
Key Tables Affected.....	11-27
import_SPED (SPED Batch).....	11-28
Design Overview .....	11-28
Scheduling Constraints .....	11-28
Restart/Recovery .....	11-28
Key Tables Affected.....	11-28
<b>Deal Income Postings Batch</b> .....	11-29
fmdealinc (Deal Income Posting).....	11-29
Design Overview .....	11-29
Scheduling Constraints .....	11-29
Restart/Recovery .....	11-30

Key Tables Affected.....	11-30
<b>ORFM Utility Batches</b> .....	11-30
fm_async_queue_cleanup (cleanup queues from async ORFM batches).....	11-30
Design Overview .....	11-31
Scheduling Constraints .....	11-31
Restart/Recovery .....	11-31
Key Tables Affected.....	11-31
fm_batch_avg_sales_price (Tax Calculation of Legislation of MG state) .....	11-31
Design Overview .....	11-31
Scheduling Constraints .....	11-32
Restart/Recovery .....	11-32
Key Tables Affected.....	11-32
fm_batch_freclass_sql (ORFM Reclassification Process).....	11-32
Design Overview .....	11-33
Scheduling Constraints .....	11-33
Key Tables Affected.....	11-33
fm_generate_nf_stkcount (Stock Count) .....	11-33
Design Overview .....	11-34
Scheduling Constraints .....	11-34
Restart/Recovery .....	11-34
Key Tables Affected.....	11-34
fm_reprocess_nf_status_f (Reprocess NF-e with status F).....	11-34
Design Overview .....	11-34
Scheduling Constraints .....	11-35
Restart/Recovery .....	11-35
Key Tables Affected.....	11-35
fm_stock_position (Stock Position).....	11-35
Design Overview .....	11-35
Stock Owner (Retailer) - Stock Possession (Retailer).....	11-36
Stock Owner (Retailer) - Stock Possession (Third-Party).....	11-36
Scheduling Constraints .....	11-37
Restart/Recovery .....	11-37
Key Tables Affected.....	11-38
fmedinf (NF EDI Batch).....	11-38
Design Overview .....	11-38
Scheduling Constraints .....	11-38
Restart/Recovery .....	11-39
Key Tables Affected.....	11-39
fmmanifest (FM Manifest Batch) .....	11-39
Design Overview .....	11-40
Scheduling Constraints .....	11-40
Restart/Recovery .....	11-40
Key Tables Affected.....	11-40
fmprepost (Pre/Post processing).....	11-41
Design Overview .....	11-41
Scheduling Constraints .....	11-41
Restart/Recovery .....	11-42

Key Tables Affected.....	11-42
110n_ld_iindfiles (Load data into template tables for item induction).....	11-42
Design Overview .....	11-42
Scheduling Constraints .....	11-42
Restart/Recovery .....	11-42
Key Tables Affected.....	11-43
<b>Localization Purge Batch Process</b> .....	11-43
fm_batch_clear_calc_stg_t (Purge localization stage tables) .....	11-43
Design Overview .....	11-43
Scheduling Constraints .....	11-43
Restart/Recovery .....	11-43
Key Tables Affected.....	11-44
fmpurge (Purging Process Batch) .....	11-44
Design Overview .....	11-44
Scheduling Constraints .....	11-44
Restart/Recovery .....	11-44
Key Tables Affected.....	11-45
110nbrfreclsprg (Fiscal Reclassification Purge) .....	11-46
Design Overview .....	11-46
Scheduling Constraints .....	11-47
Restart/Recovery .....	11-47
Key Tables Affected.....	11-47
110npurge (Localization Purging).....	11-47
Design Overview .....	11-47
Scheduling Constraints .....	11-48
Restart/Recovery .....	11-48
Key Tables Affected.....	11-48

## 12 Configuring Security for Web Services

Transport-Level Security .....	12-2
Message-Level Security.....	12-2
Access Control Security.....	12-2
<b>Implementing Web Service Security</b> .....	12-2
Testing Web Service Security .....	12-2

## 13 Oracle Retail Invoice Receiving Layer

IRL Architecture .....	13-1
Integration Overview .....	13-2
IRL Web Services .....	13-2
IRL Data Model .....	13-3
Object Structure Overview .....	13-4
Configuration.....	13-4

## 14 Tax Burden Display at POS and Sales NF

Overview.....	14-1
Tax Burden Induction Framework .....	14-1

<b>Manual Upload</b> .....	14-2
<b>Template Maintenance</b> .....	14-2
Template Header.....	14-2
Worksheet Definition.....	14-3
Column Definition .....	14-3
Spreadsheet Formatting .....	14-4
Template Definition Examples.....	14-5
IBPT Source Template Example.....	14-5

## **A Understanding RTIL WorkManager Settings**

Configuring RTIL WorkManager .....	A-1
------------------------------------	-----

## **B ORFM and Brazil Configuration Checklist**

<b>Setting up ORFM</b> .....	B-1
Setting up RMS (Brazil-On) System Options .....	B-1
Setting up Set-Of-Books ID.....	B-2
Setting up General Ledger .....	B-3
Setting up GL Account.....	B-3
Setting up Chart of Account.....	B-3
Setting up GL Cross-Reference .....	B-3
System Setup.....	B-4
System Options .....	B-4
Fiscal Number Setup: .....	B-4
Tolerances .....	B-5
Commerce Anywhere.....	B-5
Setting up SIM System Options for ORFM .....	B-5
Setting up RWMS System Options for ORFM.....	B-6
Configuring Printer.....	B-6
RMS/ORFM.....	B-6
RWMS.....	B-7
SIM .....	B-7

---

---

## Send Us Your Comments

Oracle Retail Fiscal Management/RMS Brazil Localization Implementation Guide,  
Release 14.2

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

---

---

**Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network Web site. It contains the most current Documentation Library plus all documents revised or released recently.

---

---

Send your comments to us using the electronic mail address: [retail-doc\\_us@oracle.com](mailto:retail-doc_us@oracle.com)

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at <http://www.oracle.com>.



---

---

# Preface

The *Oracle Retail Fiscal Management/RMS Brazil Localization Implementation Guide* provides detailed information that is important when implementing Oracle Retail Fiscal Management (ORFM).

## Audience

This document is intended for the Oracle Retail Fiscal Management application integrators and implementation staff, as well as the retailer's IT personnel.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Related Documents

For more information, see the following documents in the Oracle Retail Fiscal Management Release 14.2.1 documentation set:

- *Oracle Retail Fiscal Management User Guide and Online Help*
- *Oracle Retail Fiscal Management Release Notes*

## Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

<https://support.oracle.com>

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

## Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 14.1) or a later patch release (for example, 14.1.3.1). If you are installing the base release, additional patch, and bundled hot fix releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch and bundled hot fix releases can contain critical information related to the base release, as well as information about code changes since the base release.

## Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times not be attached to a numbered software release; instead, the Oracle Retail document will simply be replaced on the Oracle Technology Network Web site, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

This process will prevent delays in making critical corrections available to customers. For the customer, it means that before you begin installation, you must verify that you have the most recent version of the Oracle Retail documentation set. Oracle Retail documentation is available on the Oracle Technology Network at the following URL:

<http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html>

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-02 is an updated version of a document with part number E123456-01.

If a more recent version of a document is available, that version supersedes all previous versions.

## Oracle Retail Documentation on the Oracle Technology Network

Oracle Retail product documentation is available on the following Web site:

<http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html>

(Data Model documents are not available through Oracle Technology Network. You can obtain them through My Oracle Support.)

## Conventions

The following text conventions are used in this document:

<b>Convention</b>	<b>Meaning</b>
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.



---

---

# Oracle Retail Fiscal Management Overview

This chapter provides an overview of Oracle Retail Fiscal Management (ORFM).

The following topics are covered in this chapter:

- [What is Oracle Retail Fiscal Management?](#)
- [ORFM Functional Assumptions](#)
- [What is Nota Fiscal \(NF\)?](#)
- [What is Nota Fiscal Eletrônica \(NF-e\)?](#)
- [What is Retail Tax Integration Layer?](#)
- [What is TaxWeb Tax Engine?](#)
- [What is Sistema Público de Escrituração Digital \(SPED\)?](#)

## What is Oracle Retail Fiscal Management?

ORFM manages Nota Fiscal (NF) processing through integration with Oracle Retail Merchandising System (RMS), Oracle Retail Warehouse Management system (RWMS), and Oracle Retail Store Inventory Management (SIM). In Brazil, before shipping any inventory out of the warehouse or store, it is mandatory to generate a Nota Fiscal to accompany the inventory movement. Similarly, prior to receiving physical inventory in a warehouse or store, it is mandatory to match the NF.

ORFM is integrated with RMS, RWMS, and SIM. Since ORFM and RMS share the same instance and database schema, ORFM can look up the RMS database tables and vice versa. For example, a purchase order (PO) or transfer created in RMS is directly accessible to ORFM while a supplier, location, or item level fiscal attribute that is set in RMS can be accessed by ORFM through direct database lookup. ORFM also integrates with SIM and RWMS using the Oracle Retail Integration Bus (RIB).

ORFM foundation data is divided into following sections:

- System Setup
- Fiscal Configuration

## System Setup

The ORFM system set up involves the following tasks:

- System Options
- Fiscal Number
- Tolerances

- Requisition Types

### System Options

Set up the ORFM system parameters, including the Default Utilization codes. The configuration settings control system behavior based on these values.

**Navigate:** From the **Main** menu, select **Fiscal Management > System Setup > System Options > Edit**. The Fiscal Management System Options window appears.

### Fiscal Number

Use the Fiscal Number to set up minimum and maximum fiscal numbers allowed at a particular store or warehouse location.

**Navigate:** From the **Main** menu, select **Fiscal Management > System Setup > Fiscal Numbers > Edit**. The Location Fiscal Number window appears.

### Tolerances

Specify the amount of cost and quantity variance allowed on a fiscal document when compared to actual requisition. You can add tolerances at the system level or at the supplier level.

**Navigate:** From the **Main** menu, select **Fiscal Management > System Setup > Tolerances > Edit**. The Tolerances window appears.

### Requisition Types

This functionality supports NF creations corresponding to various transaction types. For instance, PO, PO for Rural Producer, Transfer, Repairing, InterCompany Transfer, RTV, Return NF, Return Merchandize Authorization, Inventory Adjustments (Stock Out), Direct NF, Free Form NF, Sales NF, Customer Order Dropship, and Transfer Return NF.

## Fiscal Configuration

Fiscal Configuration involves the following:

- Fiscal Document Type
- Fiscal Utilization Code

### Fiscal Document Type

In Brazil, the movement of items need a document called fiscal document which is basically a financial document that details items, quantities, cost, retail, and taxes. Create and attach a Fiscal Doc Type (FM\_DOC\_TYPE\_UTILIZATION and FM\_FISCAL\_DOC\_TYPE) to the fiscal document.

**Navigate:** From the **Main** menu, select **Fiscal Management > Fiscal Configuration > Fiscal Document Types > Edit**. The Fiscal Document Type window appears.

### Fiscal Utilization Code

Use the Fiscal Utilization Code to identify the type of fiscal operation that is being carried out by fiscal document.

**Navigate:** From the **Main** menu, select **Fiscal Management > Fiscal Configuration > Fiscal Utilization > Edit**. The Fiscal Utilization Setup window appears. Click **Add** and enter fields such as Utilization ID, Description, Requisition Type, Nature of Operation,

and Mode. Click **Options** menu to enter details for Doc Types, Parameters, and Reason Codes. Click **OK**. The Fiscal Utilization Setup window closes. The nature of operation codes list is exported by Tax Rules.

#### **Fiscal Utilization - Document Type**

Use this option to associate the relationship between Fiscal Document Type and Utilization Code.

#### **Fiscal Utilization - Reason Code**

Use this option to set up reason codes for inventory adjustments and repairing operations.

## **ORFM Functional Assumptions**

For Brazil localized users, some of the RMS functionalities are not restricted by the system. Oracle Retail assumes that you will not use RMS specific functionality. For example, the RTV and Inventory Adjustment RMS features are not available for Brazil localized users.

ORFM/RMS Brazil localization has the following functional assumptions:

- When you install ORFM, the RMS system indicator, AUTO\_RCV\_STORE, available on the RMS Supplier screen must be set to 'N'.
- The Location-level security is not enabled in ORFM.
- For EDI NF, if there are observations related to recoverable Imposto sobre Circulação de Mercadorias e Prestação de Serviços (ICMS) and Imposto sobre Circulação de Mercadorias e prestação de Serviços-Substituição Tributária (ICMS-ST), they should appear as different fields. The system cannot parse the observations.
- During discrepancy identification and resolution, the system always assumes the cost components on the Nota Fiscal are correct.
- During discrepancy identification, the system compares discounted cost on the purchase order and discounted cost on the NF. Unit cost (non-discounted) and the discount values are not compared separately.
- For triangulation purchase order flow, the NFs from both the suppliers should match in cost and quantity and should be entered in the system at the same time.
- The taxes on the main NF and the delivery NF (complementary NF for triangulation) are mutually exclusive.
- The system generates correction documents for both main and delivery suppliers.
- If the purchase NF has taxes such as Imposto sobre Produtos Industrializados (IPI) and ICMS-ST that are added to the NF total, these taxes are embedded in the return to vendor (RTV) cost on the RTV NF.
- For triangulation purchase order flow, both the delivery and main supplier should be known when the purchase order is created.
- If the stock being returned from a location was received at that location through a transfer, the system cannot track the last purchase NF and hence the RTV happens at Weighted Average Cost (WAC).

## What is Nota Fiscal (NF)?

In Brazil, all movements of products from one location to another, or from a supplier to the retailer's location and vice versa, must be accompanied by a fiscal document called Nota Fiscal (NF). This document contains:

- All commercial transaction information such as, items, quantities, costs and retail values. It can be compared to a bill of lading (BOL) because it has the items and quantities.
- All financial information such as the cost, so it can be compared to the invoice too.
- All the taxation and fiscal information, such as tax codes, tax values, Código Fiscal de Operações e Prestações (CFOP), Cadastro Nacional de Pessoas Jurídicas (CNPJ), and other fiscal attributes used by the government to verify taxes for merchandise in transit.

The management and processing of this document is closely linked to the business process flows of receiving, shipping, and all types of transactions that leverage the inventory.

Once NF is created, it is validated and submitted for receipt. Once items are received, receipt information flows to Oracle Retail Management System (RMS). NF is validated against this receipt information and approved if there are no discrepancies. Possible discrepancies include cost, quantity, and tax. If there is any discrepancy in receipt it needs to be validated as either system purchase order (PO) or NF. Inventory and transaction updates are done in RMS only after NF is approved.

## What is Nota Fiscal Eletrônica (NF-e)?

Nota Fiscal Eletrônica (NF-e) is a Brazilian government project to implement a national model of electronic fiscal documentation to replace the system of issuing the fiscal documents in paper. The virtual document will have juridical validity guaranteed by the digital signature of the issuer. It will simplify the fiscal obligations of the taxpayers and will allow the follow-up of the commercial operations by the tax authority.

The NF-e issuer will generate an electronic file with all NF information in a more detailed level than the regular NF. This file must be digitally signed to guarantee the integrity of the data and the authorship of the issuer. This electronic file that corresponds to the NF-e is transmitted through internet to the Secretaria da Fazenda (SEFAZ) - Brazilian Tax Authority, of the origin state of the issuer. The SEFAZ provides a pre-validation of the file and returns a receiving protocol (Authorization for Use), that will be necessary to the traffic of the goods.

To follow the goods, a simplify graphic representation of the NF-e called "Documento Auxiliar da Nota Fiscal Eletrônica"(DANFE) will be printed in a common paper, one copy, have highlighted access key and bar code for consultation of the NF-e through internet, to facilitate the confirmation and validation of information of the NF-e by the fiscal units.

The DANFE is not a Nota Fiscal and does not replace the NF. It's an auxiliary document of NF-e. It has the access code of the NF-e, which allows its owner to confirm the real existence of the NF-e in the SEFAZ web site.

## What is Retail Tax Integration Layer?

The Retail Tax Integration Layer (RTIL) is implemented as a Java Enterprise Edition application that exposes a Retail Tax Data Model API for calculating tax and hosts the associated tax service provider adapter(s). This layer forms the conduit between the

Oracle Retail applications and the tax service provider. The RTIL is responsible for the assembling and disassembling of the vendor specific data model to retail tax data model based on the configuration. RTIL is envisioned to host vendor-specific connectors which can communicate to the external third-party services with various protocols such as Enterprise JavaBeans (EJB), Java Message Service (JMS), Simple Object Access Protocol (SOAP), Hypertext Transfer Protocol (HTTP), or native Plain Old Java Object (POJO) based on the configuration. The subscribing application is not aware of the tax service provider. RTIL acts as a bridge between the subscribing application and third-party tax service provider.

## What is TaxWeb Tax Engine?

The TaxWeb Tax Engine aids retailers with multiple state operation, with a high level of complexity and large number of transactions, items and locations.

ORFM is integrated with the TaxWeb Tax Engine to address all Brazilian tax legislation with a high level of exception treatments. For all flows in ORFM that need to have tax calculations, the TaxWeb Tax Engine verifies that all taxes are applied considering the input parameters.

The following processes in ORFM/RMS use the tax calculation integration:

- Inbound Nota Fiscal validation
- Outbound Nota Fiscal issuing
- PO tax breakdown
- Item creation
- Refresh tax
- Item fiscal reclassification

## What is Sistema Público de Escrituração Digital (SPED)?

Sistema Público de Escrituração Digital (SPED) or Public System of Digital Bookkeeping is the result of several efforts from the Brazilian government that unifies the activities of receiving, validating, storing and authenticating books and documents that integrate accounting and tax bookkeeping. It is based on a digital file that is transmitted periodically to the government through the internet informing all operations and foundation data that can leverage taxes counting as ICMS and IPI. Similar to the NF-e, the file is digitally signed through specific programs that validate its format and content.

In RFM, a set of tables and views provides all necessary data about all fiscal documents to be integrated through an interface to the Fiscal Authority in Oracle's fiscal partners.

More detail about SPED in RFM is provided in "[Sistema Público de Escrituração Digital \(SPED\) Solution](#)".



---

---

## Understanding ORFM Implementation Checklists

This chapter discusses the prerequisites required for implementing ORFM. It also provides a checklist for standard implementation and for legacy systems.

The following topics are covered in this chapter:

- [Prerequisites](#)
- [Implementation Checklist for Legacy Systems](#)
- [Implementation Checklist for Standard Systems](#)

### Prerequisites

Ensure the following prerequisites are met before you use the ORFM application:

- If Taxweb is the Tax Engine, then make sure RMS, ORFM, RTIL, and TaxWeb Tax Engine are installed successfully.
- In case ORFM is integrated with SIM and RWMS, verify that the RIB adapters are up and running. RIB adapters ensure interface of schedule and shipment details between ORFM and RWMS/SIM. For more information, see the Oracle Retail Integration Bus documentation set, including Integration Guide.
- Set up the Localization Flexible Attributes Setup (LFAS) for a set of books, transfer entity, country, store, warehouse, supplier, outside locations, external finishers, and for the item-country combination. Localization Flex Attributes facilitate linking of fiscal attributes to respective entities. For more information, see the *Oracle Retail Merchandising System with Brazil Localization Installation Guide*.

### Implementation Checklist for Legacy Systems

Use the following checklist before you implement ORFM in a legacy environment:

1. Run the data conversion script to load the data from the legacy system. For more information, see the *Oracle Retail Merchandising Data Conversion Operations Guide*.
2. Ensure that the following are populated with correct values before making the TaxWeb Tax Engine calls. The script `transform_br_future_cost.ksh` populates the FUTURE\_COST table if there is no record in future cost table.
  - FUTURE\_COST table
  - CURRENCY\_RATES table are populated with the EFFECTIVE\_DATE prior to the transaction date

3. Refresh the MV\_CURRENCY\_CONVERSION\_RATES materialized view.
4. Run the scripts manually in the following order as shown in [Table 2-1](#).

**Table 2-1 Shell Scripts**

Script	Function
l10n_br_seed_extax_setup_cost.ksh	This module reviews all the valid item/supplier/origin country/location combinations in RMS. It determines the unique fiscal attribute groups that cover all the item/supplier/origin country/location combinations and puts the groups on the stage tables, L10N_BR_T_EXTAX_STG_COST (for TaxWeb).
l10n_br_seed_extax_process_cost.ksh	This module picks up the staged groups placed on L10N_BR_T_EXTAX_STG_COST (for TaxWeb) by l10n_br_seed_extax_setup_cost.ksh and calls the external tax provider with them. The results of these calls are placed on the L10N_BR_T_EXTAX_RES_COST and L10N_BR_T_EXTAX_RES_COST_DET (for TaxWeb) tables.
l10n_br_seed_extax_finish_cost.ksh	This module picks up the group level tax call results that l10n_br_seed_extax_process_cost.ksh placed on L10N_BR_T_EXTAX_RES_COST and L10N_BR_T_EXTAX_RES_COST_DET (for TaxWeb) and explodes them back to the item/supplier/origin country/location level. It then uses the item/supplier/origin country/location level information to write tax information to ITEM_SUPP_COUNTRY, ITEM_SUPP_COUNTRY_LOC, FUTURE_COST, ITEM_COST_HEAD, and ITEM_COST_DETAIL.
l10n_br_seed_extax_setup_retail.ksh	This module reviews all the valid item/location combinations in RMS. It determines the unique fiscal attribute groups that cover all the item/locations combinations and puts the groups on a stage table (L10N_BR_T_EXTAX_STG_RETAIL).
l10n_br_seed_extax_process_retail.ksh	This module picks up the staged groups placed on L10N_BR_T_EXTAX_STG_RETAIL (for TaxWeb) by l10n_br_seed_extax_setup_retail.ksh and calls the external tax provider with them. The results of these calls are placed on the L10N_BR_T_EXTAX_RES_RETAIL and L10N_BR_T_EXTAX_RES_RETAIL_DET (for TaxWeb) tables.
l10n_br_seed_extax_finish_retail.ksh	This module picks up the group level tax call results that l10n_br_seed_extax_process_retail.ksh placed on L10N_BR_T_EXTAX_RES_RETAIL and L10N_BR_T_EXTAX_RES_RETAIL_DET (for TaxWeb) and explodes them back to the item/location level. It then uses the item/location level information to write tax information to GTAX_ITEM_ROLLUP and POS_MODS_TAX_INFO.
l10n_br_seed_extax_future_cost.ksh	This module obtains a list of fiscal attributes that have tax law changes scheduled against them. It then uses this information to determine what item/supplier/origin country/locations will be affected by those scheduled tax law changes. These item/supplier/origin country/location combinations are then used to create new cost changes on the date of their tax law changes. The normal cost change processes will then take care of updating RMS with the new tax law information when necessary.
l10n_br_seed_extax_future_setup_retail.ksh	This module obtains a list of fiscal attributes that have tax law changes scheduled against them. It then uses this information to determine what item/locations will be affected by those scheduled tax law changes. It determines the unique fiscal attribute groups that cover those item/locations combinations and puts the groups on the stage tables, L10N_BR_T_EXTAX_STG_RETAIL (for TaxWeb).

**Table 2–1 Shell Scripts**

Script	Function
l10n_br_seed_extax_future_process_retail.ksh	This module picks up the staged groups placed on L10N_BR_T_EXTAX_STG_RETAIL (for TaxWeb) by l10n_br_extax_stg_retail.ksh and calls the external tax provider with them. The results of these calls are placed on the L10N_BR_T_EXTAX_RES_RETAIL and L10N_BR_T_EXTAX_RES_RETAIL_DET (for TaxWeb) tables.
l10n_br_seed_extax_future_finish_retail.ksh	This module picks up the group level tax call results that l10n_br_seed_extax_future_process_retail.ksh placed on L10N_BR_T_EXTAX_RES_RETAIL and L10N_BR_T_EXTAX_RES_RETAIL_DET (for TaxWeb) and explodes them back to the item/location level. It then uses the item/location level information to write tax information to GTAX_ITEM_ROLLUP and POS_MODS_TAX_INFO.
transform_br_country_attri_final.ksh	This script updates the columns default_po_cost, default_deal_cost, and default_cost_comp_cost to NIC and updates the column item_cost_tax_incl_ind to Y for the country BR in the country_attri table.
transform_br_upd_nic_cost_type.ksh	This script updates the unit cost to negotiated item cost (NIC) for the tables ITEM_SUPP_COUNTRY_LOC and ITEM_SUPP_COUNTRY. This script also updates the default_costing_type column of the future cost table to NIC.

## Implementation Checklist for Standard Systems

Use the following checklist to implement ORFM in a standard environment:

1. RMS Security setup for ORFM - Finalize ORFM user roles. In ORFM, you can control access only at the menu level. ORFM uses the RMS security infrastructure to restrict access. For instance, one role could be set up with access rights to edit or view NF while another role can be set up to resolve any discrepancy on NF. For more information on setting up RMS security, see the *Oracle Retail Merchandising System Operations Guide*.
2. RMS System setup - Verify the following in RMS before using the ORFM application:
  - Base Country is set as BR for Brazil.
  - RMS system option Currency Code is set as BRL.
  - RMS system options the DEFAULT\_TAX\_TYPE is set as GTAX to enable tax functionality.
  - CREDIT\_MEMO\_LEVEL – Ensure that the CREDIT\_MEMO\_LEVEL in the system options table is set at location level 'L'. This system option indicates at what level the invoicing will happen for deals.
  - RMS system options Allow Auto Shipping/Receiving at Store is not selected.
  - All tax codes in the VAT\_CODES table are exactly the same as the tax codes set up in the TaxWeb Tax Rules. This enables better tax integration with merchandising system.

For more information, refer to the *Oracle Retail RMS-ReSA User Guide for Brazil Localization*.

3. Verify the following setup in RMS:

- The LOCALIZATION\_IND indicator in the COUNTRY\_ATTRIB table is set to Y for Brazil.
  - The default location is set up for Brazil using the COUNTRY\_ATTRIB table. It is defined you create the first location and is most likely to be set as the location for which most of the purchases occur.
  - In the COUNTRY\_ATTRIB table, ensure the ITEM\_COST\_TAX\_INCL\_IND parameter is set to Y with the DEFAULT\_DEAL\_COST and the DEFAULT\_COST\_COMP\_COST DEFAULT\_PO\_COST parameter to (NIC).
  - The Flex field values are set to country BR. For the country BR, the fiscal attribute value should be 105.
  - The Flex field values for all entities (Store, Warehouse, Supplier/Site, Partner, Outside Location, Transfer Entity, Set of Books) are set up appropriately.
4. RTIL URL and Server Configure – Capture valid RTIL URL details based on where it is deployed. Update the RMS RETAIL\_SERVICE\_REPORT\_URL table with the URL for RS\_CODE of RTIL. Supply a timeout value (the unit is in milliseconds) in the TIMEOUT column. You can arrive at this value in conjunction with the RIB timeouts configured at the customer premise. This value should be less than the configured RIB timeout.
  5. Fiscal Information from the TaxWeb Tax Engine- Execute the fiscal download batch to obtain all necessary fiscal attributes provided by TaxWeb Tax Engine. The fiscal attributes are associated with master entities such as locations and suppliers in RMS. It is also used during transaction processing in ORFM. Verify that following fiscal attributes tables are populated in the system:
    - L10N\_BR\_NCM\_CODES
    - L10N\_BR\_NCM\_CHAR\_CODES
    - L10N\_BR\_NCM\_PAUTA\_CODES
    - L10N\_BR\_NCM\_IPI\_CODES
    - L10N\_BR\_NCM\_CEST\_CODES
    - L10N\_BR\_FEDERAL\_SVC\_CODES
    - L10N\_BR\_CNAE\_CODES
    - L10N\_BR\_UNIFIED\_SVC\_CODES
    - FM\_NOP
  6. Utilization Codes setup – Set up the utilization codes for different transaction types in ORFM. Utilization codes control the NF processing logic. The utilization codes have to be captured for every inventory transaction. The utilization code parameters control the application behavior. You need the NOP codes for setting up utilization codes.
 

More detail about Utilization Code setup, see it in *Oracle Retail Fiscal Management User Guide*.
  7. ORFM System Configuration Table – All ORFM system parameters are located in FM\_SYSTEM\_OPTIONS table. These system parameters define some system behaviors that must be informed to ensure alignment between ORFM features and customer's business requirements.
  8. A list with all ORFM System Options is in *Oracle Retail Fiscal Management User Guide*. Tolerance setup - Set up tolerances at cost, quantity (percentage or value set for system/supplier) and calculation level (for rounding or truncation) in FM\_

TOLERANCES table. During fiscal processing, the entry NF values are compared with the PO details for variances; tolerances facilitate identification of only those discrepancies which fall over and above applicable tolerance levels.

9. ORFM Tax setup – Enter the tax codes as in the TaxWeb Tax Rules in the VAT Codes Maintenance form of RMS. Tax details have to be inserted manually in the ORFM FM\_TAX\_CODES table, because there is no user interface in the ORFM application for inserting the tax details. The value added tax (VAT) codes in ORFM must be similar to the data set up in VAT codes in RMS. Also, ensure that the tax code/VAT codes set up in RMS and ORFM are as available in the TaxWeb Tax Rules.
10. Fiscal Document Numbering Sequence setup – Set up the fiscal document number sequencing details through the Location Fiscal Number form in ORFM. Fiscal document number sequence set up enables you to capture details on sequence series for each supported requisition type. During NF generation, ORFM generates and associates a unique sequence number to the NF based on its requisition type and related sequence series.
11. Brazil Localization setup – Verify that the L10N\_PKG\_CONFIG table for decoupling contains entries for Brazil before any transactions happen within ORFM. These entries enable Brazil localization related features.
12. Ensure access to the following forms (accessed through RMS forms) are denied through Security settings for Brazil localized users when Brazil localization is ON:
  - Receiver Cost Adjustment (RECCTADJ)
  - Receiver Unit Adjustment (RECUTADJ)
  - Average Cost Adjustment (AVCSTADJ)
  - Inventory Adjustment by Item (INVADJSK)
  - Inventory Adjustment by Location (INVADJST)

---

---

**Note:** Only RMS supports the multi-currency functionality.

---

---



## Setting up RMS for ORFM and Brazil Localization

This chapter describes the processes you need to prepare RMS to use ORFM and Brazil localization.

### Decoupling RMS and ORFM Batch

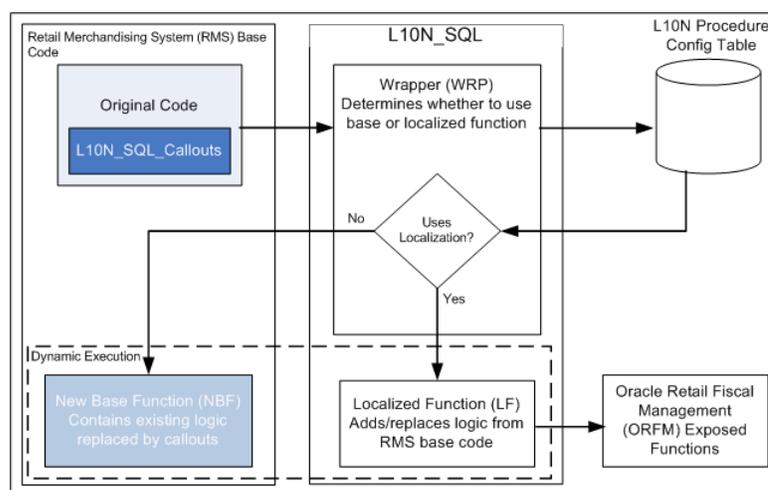
Identical processes that are performed differently in RMS and ORFM have been separated using a process called decoupling. This allows installation of RMS without installing any (or a minimal portion) of the ORFM system. This means that direct reads and writes to the ORFM tables and views from RMS are avoided.

All processes involving localization are encapsulated within a localization wrapper package. The decoupling between RMS and ORFM is controlled in that wrapper package.

### Dynamic Function Calls

ORFM and RMS utilize dynamic function calls to decouple. This method uses a configuration table to determine which function or procedure, either base or localized code, is dynamically executed.

**Figure 3–1** Dynamically Executed Decoupling



There are five components in the dynamic call approach:

1. **The original code** – This is the base code that does the callout to the localization layer.
2. **L10N\_SQL wrapper (WRP)** – This function decides whether to use the base code or the localized version of the code.
3. **L10N configuration table** – This table contains the procedure name to be executed per localized instance.
4. **New base function (NBF)** – This function contains the existing logic from the original code replaced by callouts to the localization layer.
5. **Localized function (LF)** – This contains the necessary procedures to carry out the localization tasks. This may contain actual calls to ORFM functions.

## PL/SQL Package Calls

Callouts are made in the original base code where localized codes are introduced. There can be more than one callout in a base function, and every callout points to the localization layer.

In scenarios in which code needs to be replaced, or the base code block is not applicable when localization is enabled, a New Base Function (NBF) exists to contain this code. The localized functionality is moved to a Localized Function (LF).

The localization layer contains the WRAPPER function (WRP) - EXEC\_FUNCTION, which is called by the original base code. This wrapper uses a L10N configuration table to determine if the New Base Function (NBF) or a Localized Function (LF) needs to be executed. The call to these functions is done dynamically. The configuration table may contain different function references for different localizations.

Because the calls to the functions are dynamic and the parameters passed in and out may vary depending on the function called, the PL/SQL Type Inheritance approach is used to accomplish this.

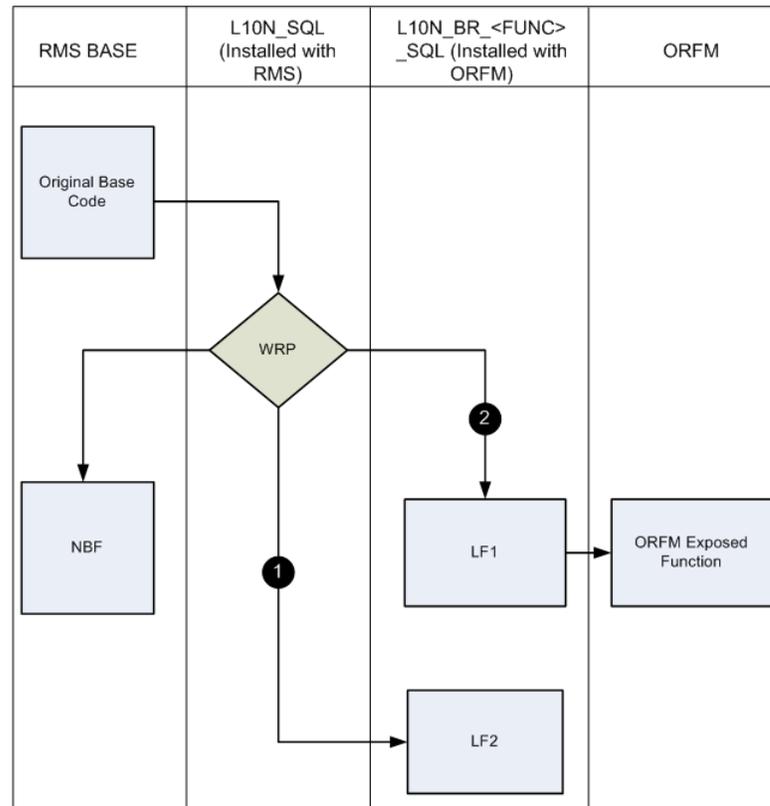
### PL/SQL Type Inheritance

In this approach, the in/out parameters to be passed to functions should be encapsulated in object types. The object types follow PL/SQL Type Inheritance (a supertype is created from which several subtypes will be derived). The supertype is a generic object that can be used across functions. Derived subtypes contain all the attributes of the parent type (or supertype). The subtypes can also contain additional attributes.

### Localization Layer

The localization layer is made up of the following components:

- The L10N configuration table
- The L10N\_SQL package
- Country-specific localization packages (L10N\_<CN>\_SQL)

**Figure 3–2 Localization Layer**

L10N\_SQL contains only the wrapper and other functions, which are generic to localized countries.

L10N\_BR\_<FUNC>\_SQL contains only the Brazil-specific functions under <FUNC> functionality area that are called by the L10N wrapper. These functions may (flow line (2) in the diagram) or may not (flow line (1) in the diagram) call ORFM functions directly.

### Accomplishing Decoupling

**BASE INSTALL:** RMS has a dependency on L10N\_SQL because it calls the wrapper directly through a callout. When RMS is installed, an L10N configuration table is pre populated with references to the New Base Functions (NBF). Because these NBF functions have no dependency on ORFM, RMS does not need the Localized Functions (LF) in the base.

**LOCALIZATION INSTALL:** After ORFM and country-specific localization packs (L10N\_BR\_<FUNC>\_SQL, L10N\_PE\_<FUNC>\_SQL) are installed, the L10N configuration table is pre populated with the localized functions for the country (BR, PE). The wrapper now calls the functions from the country-specific packages.

---

**Note:** Dependency is created only at run time because the functions are executed dynamically.

---

### Decoupling Batch Programs

In a batch program, decoupling is accomplished in a way that, if a base and localized process must occur (forking), the base logic will reside in the NBF and the localized

process will be in the LF. A wrapper is used by the main batch program to call the LF or NBF. Only one version of the function is executed in an instance of the function call.

To decouple the functions, there are separate localized libraries for LFs per country. NBFs are declared in the main batch program.

A utility library contains all the utility functions including the wrapper function. The utility function is used by the main batch program to find and retrieve the LF or NBF it has to execute. The LF pointers are organized based on the batch program that uses them, so one batch program can only use its set of LF pointers.

The NBF pointers are declared in the main batch program.

To make the parameters of the NBFs and LFs generic, a struct is used. This struct, which is called the parent struct, resides in an object library. This is the only parameter for the NBF and LFs. If additional fields are needed, the library must be modified to create a struct within the parent struct (the child struct). These structs must be populated first before passing it to the wrapper class to call the NBFs or LFs.

## Set up Item Fiscal Attributes

Item fiscal attributes are mandatory for any item creation. These attributes are referenced when the item is used in any kind of transaction data as Purchase Orders and Transfers. Item attributes include: NCM code, service indicator, NCM characteristic code, IPI exception code, pauta code, service code, federal service code, state of manufacture, pharma type list, CEST code, FUOM Unit, and FUOM Conversion Factor.

**Navigate:** From the **Main** menu, select **Items > Country > Options > Fiscal Attributes** when Country ID is 'BR'.

## Set up Fiscal Attributes for other Entities

Fiscal attributes for other entities hold information such as address, company code, and tax contributor indicators. These are necessary for tax calls and are used by the TaxWeb Tax Rules to apply appropriate rules and return the correct taxes.

**Navigate:** From the **Options** menu, select **Fiscal Attributes**. You can also navigate Fiscal attributes from the **Options** menu of Warehouses, Stores, Outside Locations, Partners, Supplier, Transfer Entity, and Set of Books.

For more information, see the *Oracle Retail Merchandising System User Guide*.

## Set up Utilization

All merchandise that is being fiscally accepted using ORFM must have the fiscal utilization, which determines the type of the business operation that the fiscal document supports.

The fiscal utilization determines the appropriate taxes involved by the retail operation, the impact on stocks and costs, and the type of information to be sent to other systems.

Utilization is associated with a Requisition Type and a Nature of Operation. This identifies the type of transaction for TaxWeb Tax Rules to calculate the appropriate taxes.

Utilization also can be associated with a particular document type. Different attributes are available to be configured for utilization which gives more granular information on the kind of document and action to be applied.

**Navigate:** From the **Main** menu, select **Fiscal Management > Fiscal Configuration > Fiscal Utilization > Edit**. The Fiscal Utilization Setup window opens.

More detail about Utilization Code setup, see it in *Oracle Retail Fiscal Management User Guide*.

## Set up Document Types

It is mandatory to have at least one fiscal document type prior to using ORFM. Each Fiscal Document Type needs to be associated with a fiscal utilization. Multiple Utilization identifiers can be associated to one document type.

**Navigate:** From the **Main** menu, select **Fiscal Management > Fiscal Configuration > Fiscal Document Types > Edit**. The Fiscal Document Type window opens.

## Set up System Options

The FM\_SYSTEM\_OPTIONS table is the control table where several parameters are stored. The system options table provides various settings such as environment type (Brazil/Non-Brazil), supplier site is on or off, and if multiple set of books are in use.

System options can be setup using the user interface or supplied through direct entries using the database. Usually system options are setup during RMS installation as part of seed data. Use the RMS user interface to update system options post installation.

You will need to setup ORFM system parameters, including the Default Utilization codes. The configuration settings control system behavior based on the values entered.

**Navigate:** From the **Main** menu, select **Fiscal Management > System Setup > System Options > Edit**. The System Options window opens.

## Set up Reason Codes

You must set up reason codes for overage and damaged in the RMS reason code master table (INV\_ADJ\_REASON) prior to using ORFM. These reason codes are interfaced to ORFM from RWMS and SIM. If these reason codes are not set up correctly, the integration to ORFM for receipt verification fails.

## Set up Partner

To use ORFM, the default partner and partner type in FM\_SYSTEM\_OPTIONS must be setup in RMS.

## Set Primary Country

To use ORFM, the Primary Country must be set to **BR**. For more information on how this can be accomplished during RMS installation, see the *Oracle Retail Merchandising System with Brazil Localization Installation Guide*.

## Set VAT Indicator

To use ORFM, VAT\_IND must be set to **Y**. For more information on how this can be accomplished during RMS installation, see the *Oracle Retail Merchandising System with Brazil Localization Installation Guide*.

## Set Default Tax Type

To use ORFM, the Default Tax Type must be set to **GTAX**. For more information on how this can be accomplished during RMS installation, see the *Oracle Retail Merchandising System with Brazil Localization Installation Guide*.

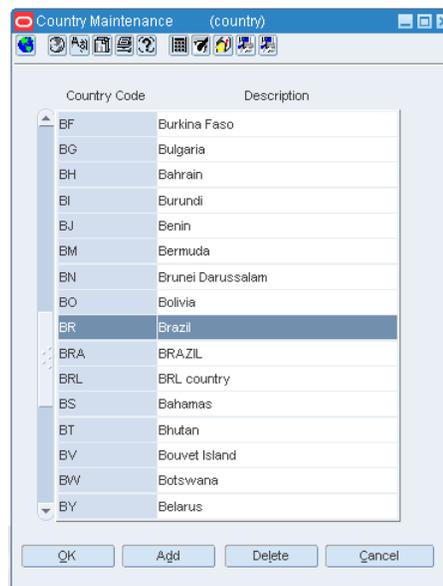
## Verify Localized Indicator

The Localized indicator check box indicates whether the Brazil patch is installed. It is auto-checked by the system if the Brazil patch is installed.

Use the following procedure to verify the Localized Indicator:

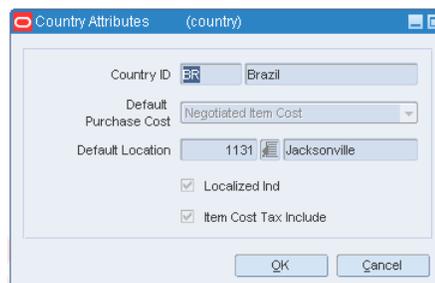
1. From the **RMS Start** menu, select **Control > Setup**. In the Content of Setup window, select **Country > Edit**. The Country Maintenance window appears.

**Figure 3–3 Country Maintenance Window**



2. Select **BR** and click the **Options** menu.
3. From the **Options** menu, select **Attributes**. The Country Attributes window opens.

**Figure 3–4 Country Attributes Window**



4. Verify that the **Localized Ind** check box is selected and click **OK**. You are returned to the Country Maintenance window.

5. Click **OK** to save your changes and close the window.

## Set up Tax Codes

Tax codes must be set up in the FM\_TAX\_CODES table. This table holds entries for all possible tax codes that can appear on a Nota Fiscal which would be applicable on any given transaction. It also holds the MATCHING\_IND (Y/N) identifier which drives the tax discrepancy identification functionality of ORFM.

More information about Discrepancy, see *Oracle Retail Fiscal Management User Guide*.

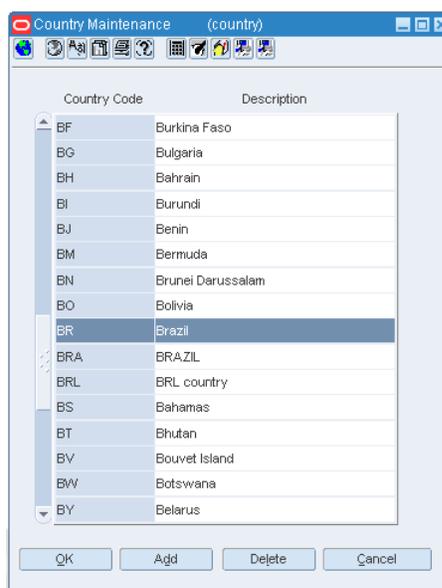
This table can only be populated by entering values directly from the back end.

## Set up Flex Field Attribute

The fiscal code of the country must be set up before using ORFM.

**Navigate:** From the Main menu, select Control > Setup > Country > Edit. The Country Maintenance window opens.

**Figure 3–5** Country Maintenance Window



1. Select **BR** and click the **Options** menu.
2. From the **Options** menu, select **Fiscal Attributes**. The Localization Flexible Attributes window opens.

**Figure 3–6 Localization Flexible Attributes Window**

The screenshot shows a window titled "Localization Flexible Attributes (110nflex)". At the top, there are two input fields: "Country Code" with the value "BR" and "Country ID" with the value "BR". Below these, there is a section for "Attribute Group" with a list box containing "Country Attributes". To the right of the list box, there are two more input fields: "Fiscal Country" with the value "BR" and "Fiscal Code" with the value "105". At the bottom of the window, there are three buttons: "OK", "Apply", and "Cancel".

3. Set the Fiscal Code to 105 and click **OK**. You are returned to the Country Maintenance window.
4. Click **OK** to save your changes and close the window.

## Set up Fiscal Document Chunk Size

The L10N\_TAX\_OBJECT\_CONFIG.THREAD\_ITEM\_LOC\_COUNT column in The L10N\_BR\_TAX\_CALL\_STAGE\_RMS table specifies the maximum number of item/locations that are contained in a Fiscal Document object. Location is defined as the 'From Entity' in the ORFM context. Logic in the ORFM integration package (L10N\_BR\_INT\_SQL) uses this value to split fiscal data into logical unit of work packets or 'chunks' that are processed concurrently by RTIL.

Chunks as used in RTIL are further discussed in [Chapter 7, "Integrating with TaxWeb Tax Rules– Retail Tax Integration Layer"](#) in section [Concurrent Processing in RTIL](#).

The value in the L10N\_TAX\_OBJECT\_CONFIG.THREAD\_ITEM\_LOC\_COUNT column defaults to 1000 through a script upon installation. Oracle Retail recommends that you set this value in conjunction with determining the configuration for RTIL's Maximum Thread Constraint and Capacity Constraint as they are intrinsically related.

## Set up FM Tran Code

ORFM Transaction codes are set up as a part of installation. A list of existing ORFM tran codes:

Tran Code	Description
100	Total merchandise cost exclusive of taxes
101	Total non-recoverable taxes
102	Total recoverable taxes
103	Total non-merchandise cost exclusive of taxes
104	Variance merchandise Cost
105	Variance recoverable taxes

<b>Tran Code</b>	<b>Description</b>
106	Variance non-recoverable taxes
107	Calculation Tolerance Cost
108	Calculation Tolerance recoverable tax
109	Calculation Tolerance non-recoverable tax
110	Return NF - total merchandise cost exclusive of taxes
111	Return NF - total non-recoverable taxes
112	Return NF - total recoverable taxes
113	Return NF - total non-merchandise cost exclusive of taxes
114	Correction NF - total merchandise cost exclusive of taxes
115	Correction NF - total non-recoverable taxes
116	Correction NF - total recoverable taxes
117	Correction NF - total non-merchandise cost exclusive of taxes
118	Complementary NF - total merchandise cost exclusive of taxes
119	Complementary NF - total non-recoverable taxes
120	Complementary NF - total non-merchandise cost exclusive of taxes
121	Complementary NF non-merchandise taxes
122	RTV NF - total merchandise cost exclusive of taxes
123	RTV NF - total non-recoverable taxes
124	RTV NF - total recoverable taxes
125	Complementary NF for Freight - total Non-merchandise cost exclusive of taxes
126	Complementary NF for Freight - total recoverable taxes
127	Complementary NF for Freight - total non-recoverable taxes
128	Complementary NF Request - total non-merchandise cost exclusive of taxes
129	Non-recoverable Taxes for transfers for source location
130	Recoverable Taxes for transfers for source location
131	Non-recoverable Taxes for transfers for destination location
132	Recoverable Taxes for transfers for destination location
133	Positive Deal Income for merchandise fixed and complex deals
134	Negative Deal Income for complex deals
135	Deal Income for non-merchandise fixed deals
136	ICMS non-recoverable Taxes when product moves into ICMSST regime
137	ICMSST And ICMSSTE taxes when product moves into ICMSST regime
138	ICMS non-recoverable Taxes when product moves out off ICMSST regime
139	ICMSST And ICMSSTE taxes when product moves out off ICMSST regime
140	Complementary NF for Cost/Tax - tax exclusive cost correction
141	Complementary NF for Cost/Tax - total recoverable taxes correction
142	Complementary NF for Cost/Tax - total non-recoverable taxes correction
143	Recoverable Taxes for NEGATIVE Inventory Adjustment

<b>Tran Code</b>	<b>Description</b>
144	Non-Recoverable Taxes for NEGATIVE Inventory Adjustment.
145	Total merchandise cost exclusive of taxes for NEGATIVE Inventory.
146	Recoverable Taxes for POSITIVE Inventory Adjustment.
147	Non-Recoverable Taxes for POSITIVE Inventory Adjustment.
148	Total merchandise cost exclusive of taxes for POSITIVE Inventory Adjustment.
150	Total de imposto FCP - ICMS-ST - retido em operação anterior.
151	Total de imposto FCP - ICMS - retido em operação anterior.
152	RTV NF - total non-merchandise cost exclusive of taxes.
161	Total merchandise retail exclusive of taxes
162	Total tax values on the transfer for customer order delivery
163	Total non-merchandise cost exclusive of taxes (Delivery charges)
164	NF-RTV Non-recoverable taxes not included in RTV NF
165	NF-RNF Non-recoverable taxes not included in Return NF
166	Tax Variance posting for RMS-RFM balance
167	Reduced ICMS Tax Value for sales or transfers done to a free zone
168	Recoverable values posting for Simples Supplier
169	Recoverable values posting for RTV to a Simples Supplier
170	Total merchandise cost for Direct Import NF exclusive of taxes
171	Total non-recoverable taxes for Direct Import NF.
172	Total recoverable taxes for Direct Import NF.
173	Total non-merchandise cost exclusive of taxes for Direct Import NF.
180	Total merchandise cost exclusive of taxes for RMA.
181	Total recoverable tax values for RMA.
182	Total non-recoverable tax values for RMA.
183	Total non-merchandise cost exclusive of taxes for RMA.
192	Value of ICMS-DIFAL to be paid for the Addressee State.
193	Value of ICMS FCP to be paid for the Addressee State.
194	Value of ICMS-DIFAL to be paid for the Sender State.
200	Total merchandise retail exclusive of taxes for sale NF future delivery customer order (backorder = Y).
201	Total tax values for sale NF future delivery customer order (backorder = Y).
205	Total merchandise retail exclusive of taxes for sale NF future delivery customer order (backorder = N).
206	Total tax values for sale NF future delivery customer order (backorder = N).
210	Total merchandise retail for shipment NF future delivery customer order.
211	Total tax values for shipment NF future delivery customer order.
212	Total non-merchandise cost exclusive of taxes for shipment NF future delivery customer order (Delivery charges).
213	Recoverable Taxes for SALES for source location (Future Delivery only).

Tran Code	Description
214	Recoverable Taxes for SALES for source location.
215	Return TSF Non-recoverable taxes not included in RTSF NF.
220	Cost exclusive of Taxes - Difference between nominal and received weight for Catch Weight items.
221	Non-recoverable Taxes - Difference between nominal and received weight for Catch Weight items.
222	Recoverable Taxes - Difference between nominal and received weight for Catch Weight items.

## Set up Set Of Books

The system administrator must set up Set of Books in FM\_SOB\_SETUP table.

## Set up Chart of Account

The system administrator must set up Chart of Accounts in FM\_COA\_SETUP table.

## Set up GL Options

The system administrator must set up the entities that are dynamic in nature in FM\_GL\_OPTIONS table.

## Dynamic Segments Set up for Chart of Accounts

The system administrator must set up dynamic segments for chart of accounts in FM\_DYNAMIC\_SEGMENT\_SETUP table. You can also achieve this through the Dynamic Segment Configuration form.

## Account Set up for Chart of Account

The system administrator must set up account setup for chart of account in FM\_ACCOUNT\_SETUP table.

## Set up GL Cross Reference

The system administrator must set up GL cross reference in FM\_GL\_CROSS\_REF table. You can also achieve this through the GL Cross reference form.

## Including Balance to RTV Balance Control

When creating RTV NF with Balance Control enabled (FM\_SYSTEM\_OPTIONS.VARIABLE = 'BALANCE\_CONTROLE\_RTV' = 'Y'), the process gets balance information from the following table (FM\_FISCAL\_DOC\_DETAIL\_BALANCE) where must contain information about PO NF received.

The FM\_FISCAL\_DOC\_DETAIL\_BALANCE table will hold information about received quantity from a PO NF in order to perform balance to Item/Supplier/Location. It should include Unit Cost from PO NF and a Quantity field to Balance Control.

## Transfer Ownership

The system can be configured to manage the moment when the transferred inventory changes ownership during transit inventory in transfers, including allocations. This moment would occur at "Time of Shipment" or "Time of Receipt".

The ownership is characterized by the definition of where the stock is in two aspects: physical and financial, that is, whether the in-transit inventory is part of the total inventory of location A or B, as well as the financial postings that will reflect, from an accounting perspective, where the inventory is.

In "Time of Shipment", when shipment occurs, the inventory is moved from the 'on hand' bucket at source to an 'in transit' bucket at destination and the financial postings happen to change ownership. At this point, this inventory already belongs to the destination location which justifies the recalculation of WAC for the transfer at that point in time. Being part of inventory at destination, it is also considered for WAC calculations in other receiving that may happen while the transfer is not yet physically received.

In "Time of Shipment", when shipment occurs, the inventory is moved from the 'on hand' bucket at the source to an 'in transit' bucket at the destination, keeping the view of in-transit currently posted for the destination location, but without posting the trancodes that would change ownership. In this case, the 'in transit' stock presented at the destination location will be taken as expected inventory to be received at some point, but not being part of destination location's total inventory from an accounting perspective. This 'expected' inventory will continue being considered in replenishment processes.

While the 'in transit' view will continue demonstrating what is expected at the destination location, it is necessary to have a new bucket created for the source location with the information of 'shipped' inventory, keeping the stock ledger position in-sync with the perpetual inventory.

With the proposed change, the in-transit bucket will continue being displayed for the destination location and considered for replenishment processing. However, it will NOT be considered for WAC updates as, in this scenario, it still belongs to the source.

In the same way, the 'shipped' bucket will be displayed for the source location but will also not be considered for WAC updates. The expectation in this case is that this is an inventory not expected to return, and hence shouldn't be accounted for WAC updates.

From an accounting perspective at source, at time of shipment, trandata will be posted, demonstrating the inventory moving from source 'on hand' bucket to source 'shipped' bucket.

## PO Receiving at a Warehouse

This behavior is controlled by system option that can change the system behavior of change ownership at shipment or at receiving.

This system option is located in the System Parameter Maintenance Window, in tab **Inventory**.

When this option is selected, it indicates that WAC updates will happen after receiving a transfer.

## PO Receiving at a Warehouse

At the time of shipment, trancodes 30/32 (for regular transfers) or 37/38 (for intercompany) are posted, considering the quantity shipped and the transfer cost. These codes represent the transfer of ownership of the inventory in a transfer.

At the time of receiving, these trancodes will be posted only at time of receiving. During the time the transfer is in-transit it is necessary to have the accounting system aware that inventory is not on-hand, but it is in the 'shipped' bucket still belonging to the source. For this specific transaction, codes exist to handle it.

- Tran Code 45 - *Data warehouse outbound transfer shipment*. This trancode is posted for the source location at the time of shipment, with the item quantity shipped and the transfer cost used (WAC of source). This trancode is informational and represents the movement of inventory from the 'on hand' to the 'shipped' bucket.
- Tran Code 46 - *Data warehouse outbound transfer shipment received*. This trancode is posted for the source location at the time of receiving with the same values posted for destination location with trancode 44. This trancode is informational and will represent the movement from 'shipped' bucket out of source location's inventory.



---

---

## Integrating ORFM

ORFM is integrated with other Oracle Retail applications including RMS, RWMS, and SIM. Because RMS and ORFM share the same instance and same database schema, ORFM can look up the RMS database tables and vice versa. A purchase order (PO) or transfer created in RMS is directly accessible to ORFM. Supplier, location or item level fiscal attributes that are set in RMS can be accessed by ORFM through direct database look up.

ORFM integrates with SIM and RWMS through RIB (for real-time integration). If Brazil localization is enabled, SIM or RWMS should not be allowed to physically receive any inventory before fiscal receiving (inbound NF Processing) is completed in ORFM. After physical receiving in SIM or RWMS, the inventory in RMS should not be updated until fiscal receipt and physical receipt comparison is done in ORFM and the discrepancy (if any) is resolved as explained above.

If Brazil localization is enabled, RMS inventory is not updated immediately when any inventory is shipped out from RWMS or SIM. The inventory movement information should be sent to ORFM for the generation of outbound NF. After approval of the outbound NF, ORFM should send the inventory updates to RMS. For more information, see the Outbound Operations Process Flow diagram in the "Overall Solution Landscape" section.

In the subsections below, each transaction with a different integration flow (other than the base application) for Brazil is detailed.

The following topic and sub topics are covered in this chapter:

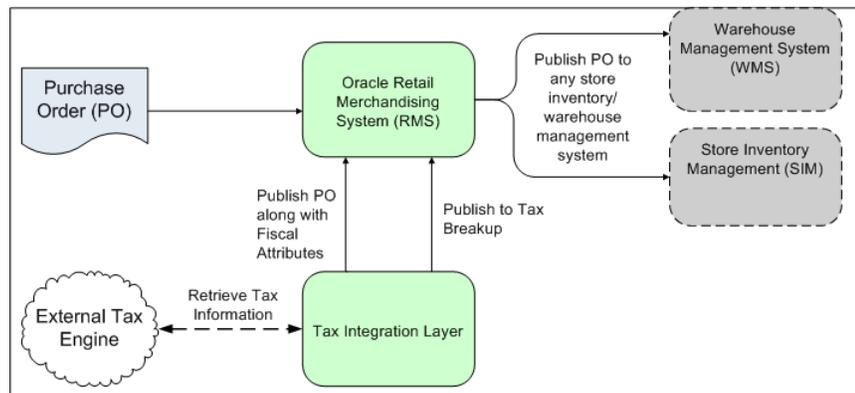
- [Purchase Order Receiving](#)
- [Transfer Shipment and Receiving](#)
- [Retail Merchandise Authorization](#)
- [Inventory Adjustment](#)
- [Finishers/Repairing Flow \(Two-Legged Transfer\)](#)
- [RIB-Based Integration](#)
- [HTTP-Based Integration for TaxWeb](#)
- [Nota Fiscal – Receiving and Issuing](#)
- [ORFM and RWMS](#)
- [ORFM and SIM](#)
- [ORFM Integration with Financial Application](#)

## Purchase Order Receiving

Purchase orders are created in RMS and communicated to Vendor; Vendor sends items back to retailer based on purchase order and generates invoice to retailer. Based on items received, invoice is adjusted by retailer and payment is made to the vendor. PO can also be created by ORSIM through direct store delivery functionality and communicated to RMS.

In Brazil, receiving without a valid PO number is not allowed because fiscal receiving must precede physical receiving. Therefore, a direct store delivery without a PO cannot be received in SIM. A PO created in RMS is sent to RWMS and SIM through RIB as per the base process flow. The PO created flows to the third-party Tax Rules through the Tax Integration Layer, along with the fiscal attributes of the suppliers, receiving locations, and the items. The third-party tax engine calculates the tax break down and sends it back to RMS through Tax Integration Layer.

**Figure 4–1 Purchase Order Receiving**




---

**Note:** RTIL is the layer for Tax Rules integration and PL/SQL is the layer for TaxWeb.

---

A purchase order goes through the following process and is published to RWMS or SIM:

1. The purchase order is created in RMS.
2. The purchase order, along with the PO's fiscal attributes, are published to Tax Integration Layer.
3. The Brazil tax information is retrieved and appended to the purchase order and sent back to Tax Integration Layer.
4. Tax Integration Layer publishes the tax breakdown of the purchase order.
5. The purchase order is published to SIM or RWMS.

---

**Note:** Steps 2, 3 and 4 depicted above are specific to Brazil localization. Steps 1 and 5 are the same as in base RMS.

---

ORFM has direct access to the PO created in RMS and the related fiscal details. This is required to validate and match the Nota Fiscal.

The following PO details are relevant to ORFM:

- PO header information
- Order supplier master data
- Order supplier site master data
- Order supplier site fiscal attributes
- Delivery supplier master data
- Delivery supplier site master data
- Delivery supplier site fiscal attributes
- Destination locations master data
- Destination location fiscal attributes
- PO detail information
- Non-merch cost details
- Item master data
- Item fiscal attributes

## **PO Receiving at a Warehouse**

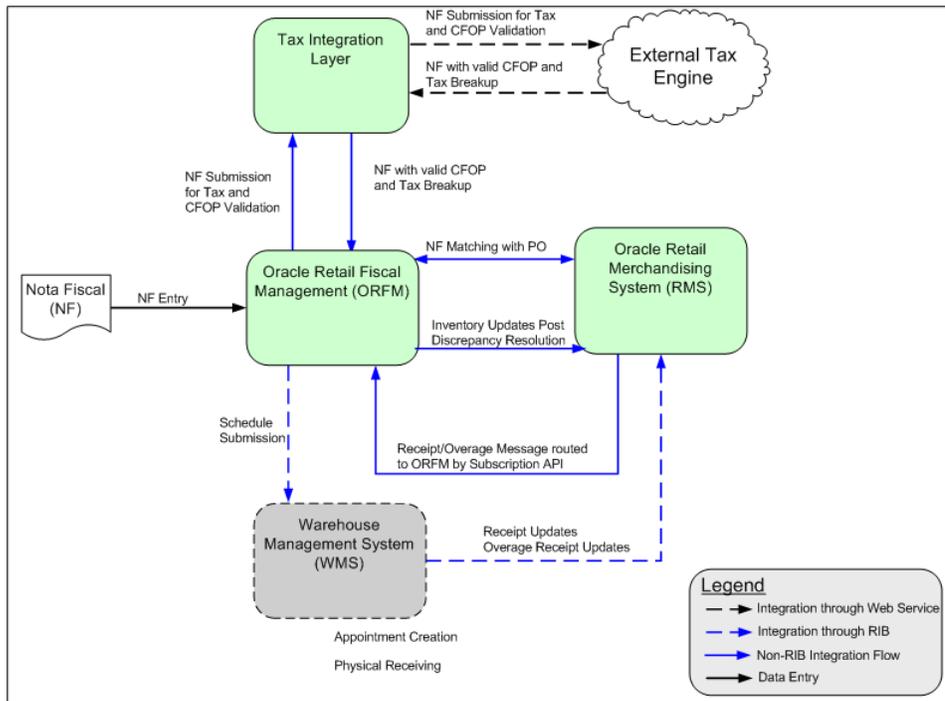
For a PO with a warehouse as the receiving location, the NF schedule status is Submitted. The message confirming that fiscal receiving is complete is published to Oracle Retail Warehouse Management System (RWMS) through the Oracle Retail Integration Bus (RIB).

The RIB message contains the following information:

- Schedule number
- PO number
- Receiving physical warehouse number
- Distinct item IDs per PO
- Consolidated quantities per item-PO
- UOM

One Nota Fiscal may contain items pertaining to several POs, and one schedule may contain several Nota Fiscal documents. It is likely that the supplier ships the same item on a PO through multiple NFs, and these NFs are part of the same schedule. If so, ORFM consolidates the item quantity by PO number before publishing it to RWMS.

Figure 4-2 PO receiving at a Warehouse



After the automatic appointment creation, the RWMS user can schedule the appointment and proceed with the physical receiving. If Brazil localization is enabled in RWMS, receiving an item that is not in the appointment or receiving any extra quantity than in the appointment is restricted. Any such overage should be received separately in RWMS as an overage receipt.

After completing the receipt, as the appointment is closed, the receipt and the overage receipt are published by RWMS through RIB. RMS consumes these messages and the Subscription API routes the messages to ORFM (in Brazil Environment). When receiving, the same receipt message is used that is used to update receipt information to RMS. Though not all fields in the receipt message are relevant to ORFM, ORFM still consumes the entire message and persists it until the RMS inventory is updated.

The Receipt Overage message is a RIB message from RWMS to ORFM.

The message contains the following information:

- Schedule Number
- ASN Number (Null for Warehouse receipts)
- Item Id
- Quantity Over-received
- UOM
- Reason Code

After ORFM consumes the Receipt and Receipt Overage messages, the discrepancy between physical and fiscal receiving is processed through the discrepancy resolution module. The correction letter and/or Return NF (for an overage receipt) are generated. No inventory updates happen in RMS for the overage receipt.

For the received items, the taxes are stripped from the unit cost to arrive at the tax-exclusive cost, additional cost components (if any) are added to calculate the actual

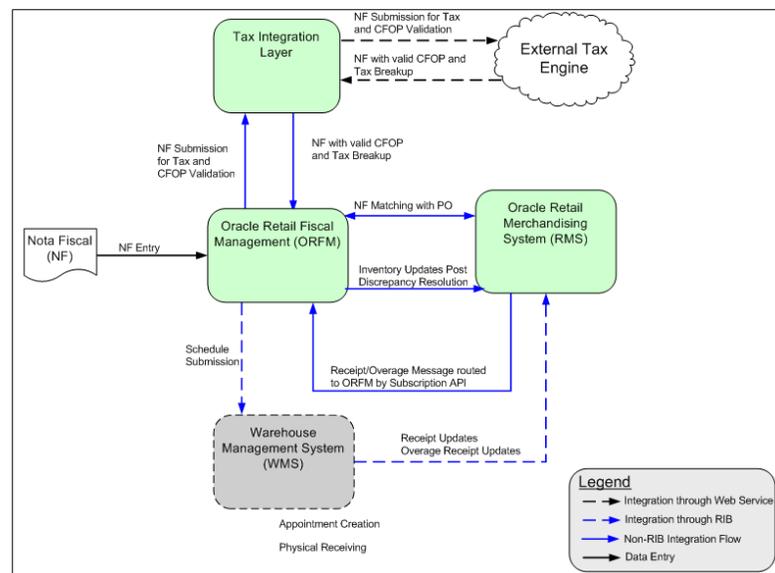
landed cost, and the inventory and WAC updates are sent to RMS along with the NF number. Other than the cost and NF, all other information regarding the shipment is taken from the receipt message received from RWMS.

## PO Receiving at a Store

For a PO with Receiving Location as a Store, if the NF schedule status is Submitted, the ASNs (ASNin) are generated and published to SIM. SIM then consumes this message to create the inbound ASN.

One NF can contain items pertaining to several POs, and one schedule can contain several NFs. It is likely that the supplier ships the same item on a PO through multiple NFs, and these NFs are part of the same schedule. If so, ORFM consolidates the item quantity by PO number before publishing it to SIM.

**Figure 4-3 PO receiving at Store**



After the ASN message is received, the SIM user can proceed with the physical receiving. For Brazil localization, PO receiving without a valid inbound ASN is not permitted in SIM. In addition, receiving an item that is not in the ASN, or receiving any extra quantity than in the ASN, is restricted.

SIM can enforce this restriction in the following ways:

- The user can continue receiving inventory beyond the expected quantity on the NF, but the extra quantity will be removed from the transaction when confirming the delivery.
- The user is not allowed to receive any units above the expected quantity.

When receipt is completed for each ASN, the receipt and the overage receipt per ASN are published to RMS through RIB. For a receipt, the same receipt message is used that is used to update receipt information to RMS. The RMS subscription API does not consume the receipt message for the Brazil environment, but it is diverted to ORFM. ORFM stores the information in a staging table until all the ASNs in a schedule are received. The schedule status is changed to Received only when all the ASNs in a schedule are received. Referential integrity between the schedule number and the ASN is maintained in ORFM. SIM has no visibility to the schedule number. Not all the fields

in the receipt message are relevant in ORFM, but ORFM still needs to consume the entire message and persist it until RMS inventory is updated.

The Receipt Overage message is a RIB message from SIM to ORFM.

After ORFM receives the Receipt and Receipt Overage messages, the discrepancy between physical and fiscal receiving is processed through the discrepancy resolution module. As required, the correction letter and/or Return NF (for an overage receipt) are generated. No information about the overage receipt flows to RMS from ORFM.

For the received items, the taxes are stripped off from the unit cost to arrive at the tax-exclusive cost. Additional cost components (if any) are added to calculate the actual landed cost. ORFM then calls RMS to update inventory and WAC. Other than the cost and the NF number, all other information regarding the shipment is taken from the receipt message that SIM has published.

## Transfer Shipment and Receiving

A transfer (inter-company as well as intra-company) or allocation created in RMS is sent to RWMS and SIM through RIB.

ORFM has direct access to the transfer and the related fiscal details.

## Transfers

Transfers are created to move items from one location (source) to other (destination). Location could be store or warehouse.

---



---

**Note:** The following section uses warehouse-to-warehouse transfers to illustrate the transfer process. Other transfer processes, including warehouse-to-store, and store-to-store are similar from an ORFM perspective.

---

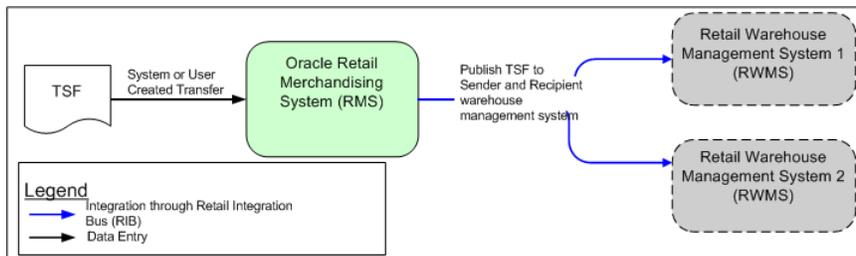


---

### Transfer Creation

If a warehouse-to-warehouse transfer (or allocation) is initiated in RMS, the transfer request flows to RWMS (Sending Location and the Receiving Location) through RIB.

**Figure 4–4 Warehouse to Warehouse Transfer Creation**




---



---

**Note:** The sender warehouse and the recipient warehouse may run on the same instance or different instances of RWMS. If both warehouses run on the same instance of RWMS, RWMS-1 and RWMS-2 in the above image would be the same.

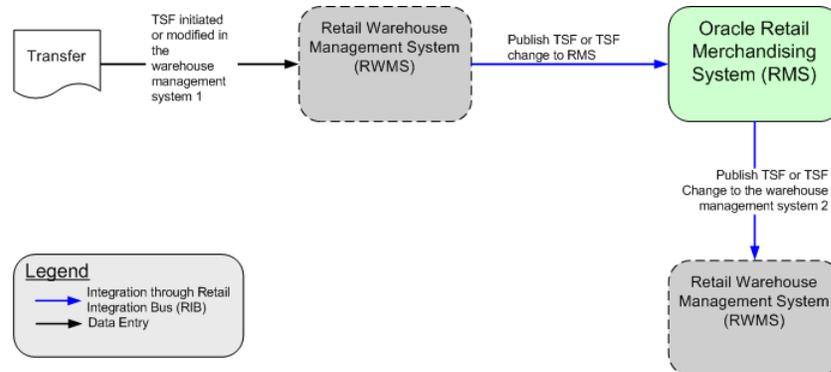
---



---

If a warehouse-to-warehouse transfer is initiated in RWMS, it flows to RMS and the other RWMS through RIB. If the RMS-initiated transfer is modified in RWMS-1, the modifications flow back to RMS and the other RWMS.

**Figure 4-5 Warehouse to Warehouse Transfer**

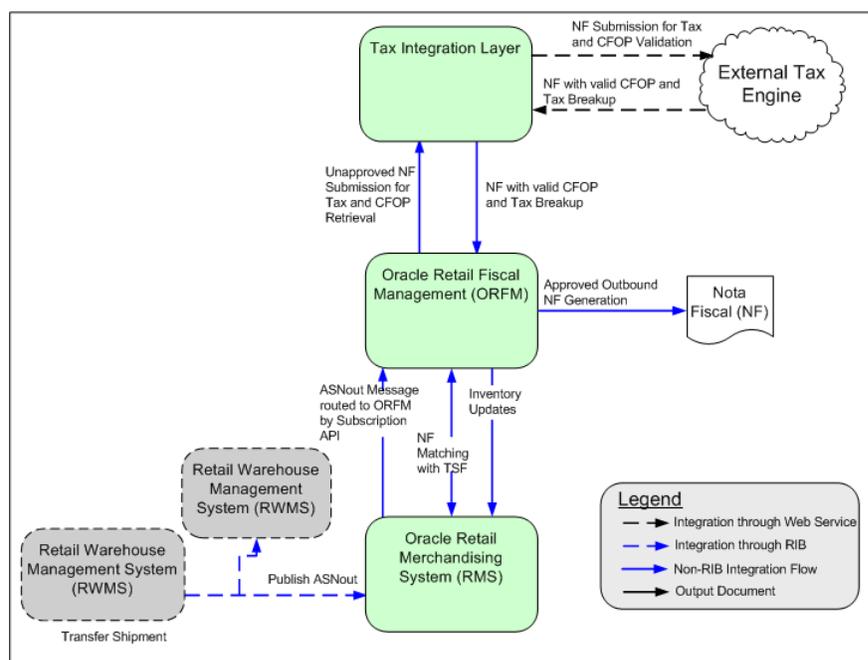


**Transfer Shipment at Source**

In Brazil, each inventory movement (shipment) should be accompanied by a fiscal document (NF). When the shipment goes out of the source warehouse, an NF is generated and approved. When RWMS ships the transfer out, ORFM is informed about the ASNOut details so that the outbound NF can be generated. On approval of the outbound NF, the on-hand inventory of the source warehouse and in-transit inventory of the destination warehouse are updated in RMS.

The ASNOut should be sent to the destination warehouse too, so that the destination warehouse generates the appointment and is prepared for the receiving.

**Figure 4-6 Transfer Shipment at Source**



When the transfer is picked and shipped in RWMS-1 at the source warehouse, ASNout is published and is subscribed to by RMS and RWMS of the destination warehouse simultaneously. In the Brazil environment, RMS routes the ASN message to ORFM. On consuming the ASN, ORFM generates the outbound schedules and outbound Nota Fiscal documents using the ASNout information. The tax information and the CFOPs are published on the NFs. NFs are validated before approval. On NF approval, the ASNout information is used to update the RMS inventory subsequently. The on-hand inventory of the source warehouse is decreased and the in-transit inventory for the destination warehouse is increased.

### **Transfer Receiving at Destination**

When the shipment arrives at the destination warehouse, the user is expected to create an inbound schedule using the outbound schedule. There should be an inbound NF for each outbound NF in the schedule, and a one-to-one relationship must exist between the outbound schedule and inbound schedule, as well as outbound NFs and inbound NFs. As the schedule is entered and validated (retrieve appropriate inbound CFOPs through TaxWeb Tax Rules), the schedule can be submitted to RWMS-2 for receiving. Because RWMS-2 already has the ASN information, it is enough that the message published to RWMS-2 contains just the schedule number and the ASN numbers grouped under the same schedule. To avoid introducing a new RIB message, the schedule is submitted with the entire ASN (referred to as secondary ASN). RWMS can ignore the other information in the message and can just refer to the ASN numbers. The schedule number can be null, as in the case of publishing an ASN to SIM.

RWMS should create one appointment per schedule and combine all the ASNs for that schedule under the same appointment. After scheduling the appointment, the RWMS user can proceed with the physical receiving. It is possible that the user finds a discrepancy between the NF and the physical receipt. Any overage is consumed by ORFM, but no discrepancy resolution against overages is performed. In case of under-receipt, the receipt message coming from RWMS is the same as that of the ASN. In case of under-receipt in the destination warehouse, the inventory mismatch in the source warehouse should be handled manually by doing an inventory adjustment. In the Brazil environment, the RMS subscription API routes the receipt, overage, and inventory adjustment messages to ORFM. If the inventory adjustment reason code is associated with an utilization code, the NF should be generated in ORFM.

On receiving the receipt message from RWMS-2, the inventory updates are done in RMS. The in-transit inventory at the destination location is reduced and on-hand inventory is increased. WAC is recalculated for the destination location. Subsequent to the inventory updates for receipts, the inventory is updated for the inventory adjustment for any under-receipt.

After physical receiving is done at RWMS-2, RWMS-2 publishes the receipt and overage receipt (if any) message to ORFM through the RMS subscription API. Though ORFM may consume this message, ORFM does not use the information for updating the inventory. Any over-receipt at RWMS is handled outside the system.

---

---

**Note:** EG transfers created directly in RWMS without prior creation of a transfer in RMS is not supported in Brazil Localization install. Only EG transfers created using stock order subscriber is supported.

---

---

## Return to Vendor (RTV) Shipments

An RTV request can be initiated in RMS (for returns from store), or in SIM and RWMS. An RTV request initiated in RMS should flow to SIM through RIB as per the existing flow.

### RTV Shipped from Warehouse

When the RTV is picked and shipped from the warehouse, an RTV Shipment message should flow from RWMS to RMS through RIB. In the Brazil flow, after consuming the message, RMS routes it to ORFM; ORFM consumes the message from RMS.

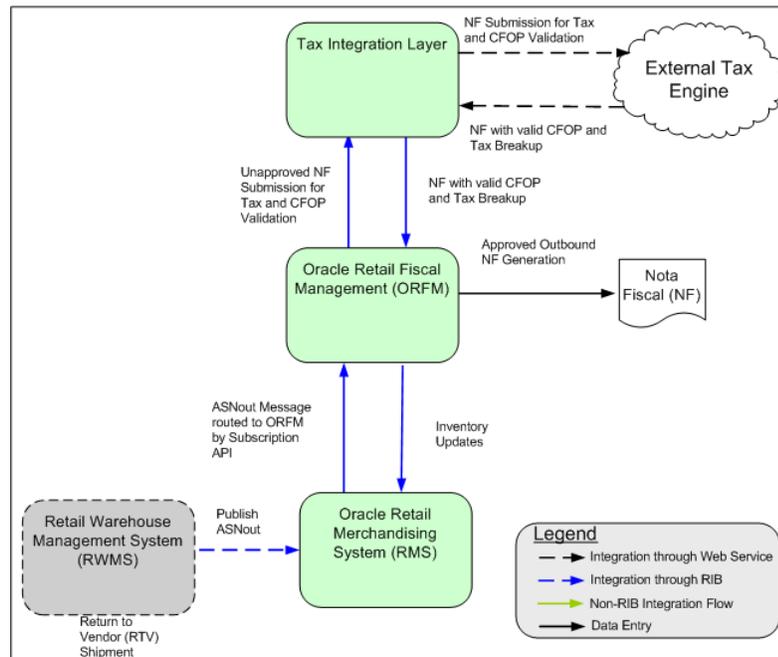
Using the information in RTV Shipment message, ORFM should generate the outbound RTV NF. If there is reference to the inbound NF (through which the stock was received), the tax information is also retrieved from the inbound NF. If no inbound NF reference is required, the system should obtain the tax information through the TaxWeb Tax Rules. The TaxWeb Tax Rules should return appropriate CFOPs too. The NF should then be auto-approved. The shipment can leave the warehouse on printing of the approved NF.

The inventory updates are done in RMS and on-hand inventory is reduced. After the RMS updates are successful, the NF is approved. The RTV cost communicated to RMS is the tax-exclusive cost of the inventory returned. While making the transaction data entry, RMS should compare the RTV cost with the WAC, and if there is a mismatch, RMS should post a cost variance appropriately.

In Brazil, any outbound shipment should have a valid approved outbound NF accompanying it. Neither RWMS nor ORFM can control the outbound trailer leaving the warehouse or store without a valid NF. Business discipline must be followed to ensure this.

If for any reason the outbound NF cannot be generated or approved, the shipment cannot leave the source location. The RTV needs to be cancelled in that case. The system currently does not handle the cancellation of an RTV post-shipment from RWMS. It needs to be handled through manual inventory adjustments in RWMS and RMS simultaneously.

Figure 4-7 RTV Shipped from Warehouse



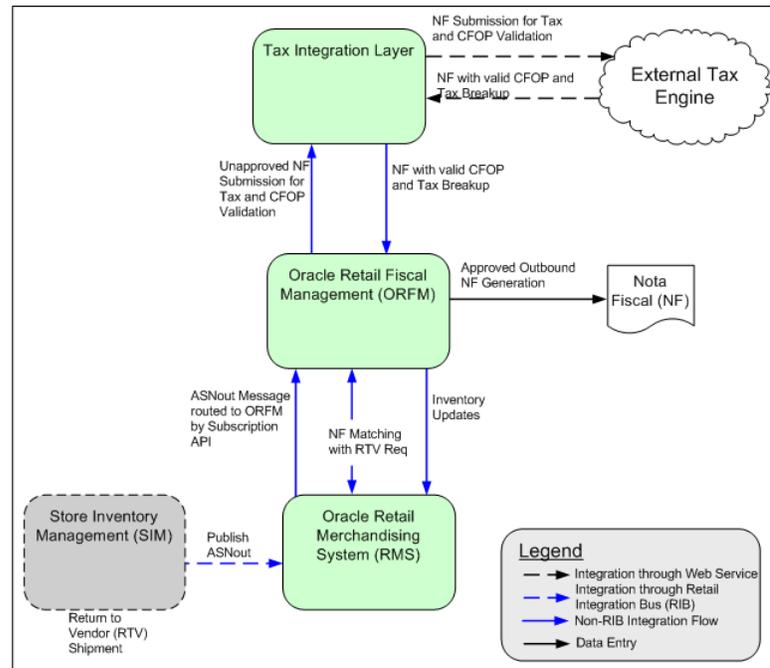
## RTV Shipped from Store

From an ORFM perspective, RTV shipments from stores are similar to RTV shipments from RWMS. When the RTV is shipped from the store, the RTV Shipment message flows from SIM to RMS through RIB. In a Brazil environment, RMS should direct the message to ORFM.

Using the information in the RTV shipments message, ORFM generates the outbound RTV NF, obtains the tax information through the TaxWeb Tax Rules, matches the NF with the requisition (transfer request in RMS, if available), and approves the NF. The shipment can leave the warehouse on printing of the approved NF.

During NF approval, the inventory updates and transaction data posting API in RMS is called. After RMS updates are successful, the NF is approved.

Figure 4–8 RTV Shipped from Store



## Balance Control for RTV

Fiscally, in the Return to Vendor operation it is assumed that the same merchandise received from supplier will be return. In Brazil, as any merchandise movement, this process requires a specific NF document to accompany the returning merchandise. This NF must obligatorily refer the NF received from the supplier for the PO, with the same tax basis and rates applied. Considering this, for tax compliance, the quantities returned must be lower or equal to the quantity received in the PO.

As government system authorize electronically a NF document, if no reference PO NF is specified in the RTV NF, it will be rejected for authorization. And as fiscally a return operation must fiscally reflect the original operation, if the same tax basis and tax rates are not applied, retailers will be subject to penalties.

When Balance Control for RTV is Enabled (`FM_SYSTEM_OPTIONS.VARIABLE = 'BALANCE_CONTROLE_RTV' = 'Y'`), the process to create an RTV NF consider the received quantity from a PO NF to create as many as lines/references to fullfil the RTV message quantity.

In the RTV NF creation, if the returned quantity to vendor is not enough to meet the balance from the most recent PO NF, the process will search the next previous NF(s), inserting new lines/references NF up to reach the total quantity returned to vendor.

The same RTV item can be inserted multiple times considering the following information from PO NF with different fiscal documents, Unit Cost, Triangulation information.

After RTV NF created, use reference from each item to retrieve taxes from PO NF and calculate them in proportion to quantity returned.

## Retail Merchandise Authorization

The third-party management system sends Retail Merchandise Authorization (RMA) information to ORFM. ORFM consumes this information and creates an inbound NF in Pending for Receiving status which informs the user that NF is created and waiting for the goods to be received.

Once the goods are received at warehouse, RWMS sends the returns receipt message to the third party. The third party sends the Processed RMA information to ORFM and at the same time it sends the RTLOG file to ReSA for inventory updates.

The user can cancel the NF in ORFM, if the goods are not received at RWMS.

Following are the assumptions:

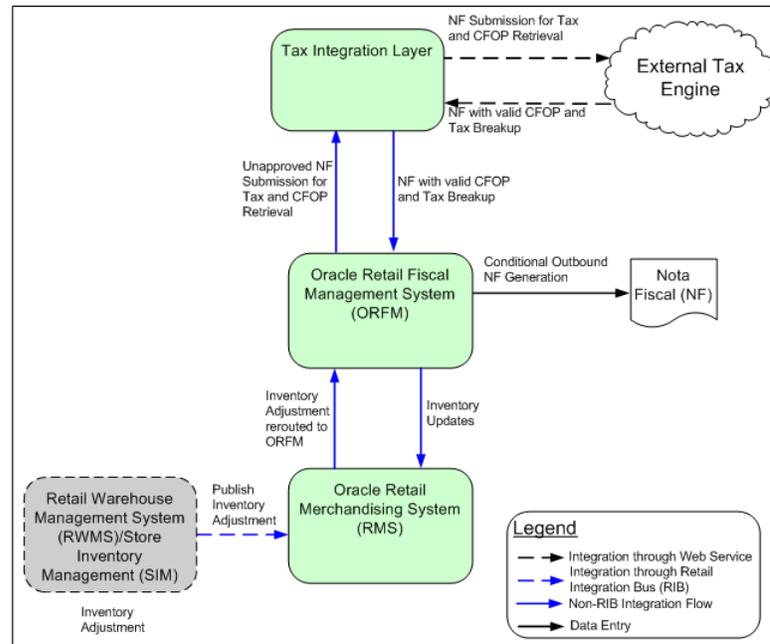
- RMA is supported only in RWMS since Return Receiving process is not supported by SIM
- You need to attach Virtual Store to the Warehouse in RMS so that RTLOG files can be created for that virtual store
- RTLOG files is generated by Third party order management system and sent directly to ReSA after the completion of RMA
- You need to create RMA manually in RWMS during the physical receiving with NF, since there is no integration between ORFM and RWMS for RMAs
- Third party order management application sends the costing information of the items related to corresponding RMA to ORFM
- Third party Order Management application validates the customer ID along with its fiscal details before sending to ORFM
- Only the processed status is sent, when the RMA processing is completed in Third party Order Management application
- There is no Discrepancy Identification and Resolution process for RMA

## Inventory Adjustment

In Brazil, the inventory adjustment may also require NF generation, depending on the reason for which the inventory was adjusted. The same inventory adjustment reason codes defined in RMS are available in RWMS and SIM. In ORFM, the utilization codes are associated with the reason codes. It is not mandatory that each reason code is associated with some utilization code, but if the reason code is associated with the utilization code, an outbound NF is generated.

Because the inventory adjustment is done in SIM or RWMS, an Inventory Adjustment message should be published by SIM or RWMS. This message requires no changes to support Brazil specific flows. The message is consumed by RMS, and the subscription API routes it to ORFM in the Brazil environment. If the reason code associated with the inventory adjustment requires NF generation, an outbound NF is generated. After the NF validation and approval, the inventory in RMS is updated. If the reason code associated with the inventory adjustment does not require NF generation, using the same information received in the Inventory Adjustment message, RMS inventory is updated.

Figure 4–9 Inventory Adjustment



## Finishers/Repairing Flow (Two-Legged Transfer)

A retailer can send goods out to an external finisher or supplier for either finishing work such as printing, dying, or embroidery, or repair in the case of damaged goods. After the job is done, the finisher or supplier returns the goods to the same or a different location. While the goods are out of warehouse or store for repair or finishing, RMS needs to account for the inventory as the retailer's inventory.

This flow needs to be handled through two-legged transfers. The first leg of the transfer corresponds to the inventory being sent from the retailer's location (warehouse or store) to the finisher's location. The second leg corresponds to the receiving the finished goods (same or different SKU) or repaired goods back from the finisher at the retailer's location. Because the finisher's location is an outside location, the retailer is not responsible for the receiving or shipping at those locations.

### Two-Legged Transfer Shipping from RWMS

When goods need to be sent to external finishers for finishing work, a two-legged transfer must be initiated in RMS, with transfer type as manual requisition. The first leg of the transfer (warehouse to finisher) flows to RWMS through RIB. RWMS persists it as a regular stock order. RWMS needs to do no special processing for such stock orders.

When goods need to be sent to external finishers or suppliers for repair, a two-legged transfer can be initiated in RMS, with the transfer type as manual requisition and the context type as repair. The first leg of the transfer (warehouse to finisher) flows to RWMS through RIB. RWMS stores it as a stock order with the type set to Repairing. All Repairing types of stock orders need special processing in RWMS. A regular wave should not select these stock orders for picking. These stock orders are picked manually, and only the inventory associated with appropriate trouble code can be picked up for repairing.

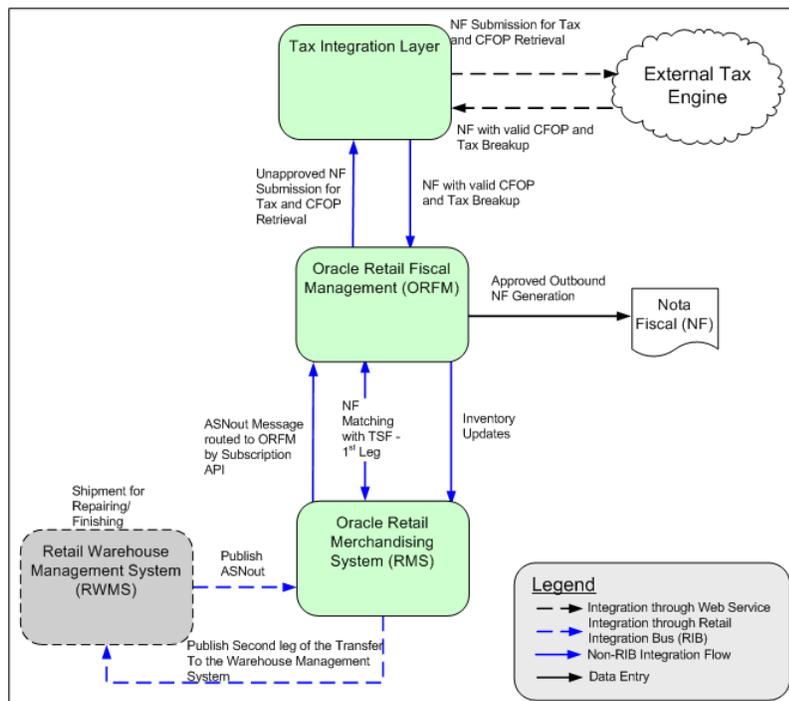
RWMS, as the physical owner of the inventory, has better visibility to the goods that need to be repaired. Therefore, a repairing stock order can be initiated in RWMS as well. As a stock order with the type of Repairing is created in RWMS, it flows to RMS through RIB. RMS, on consuming the message, creates a two-legged transfer with a context type of Repairing with the final location as the same origin warehouse.

As the required stock is picked and shipped at RWMS (for the external finisher's location), the ASNout message is published. ORFM consumes this message through the RMS subscription API and generates an outbound NF with Repairing as the requisition type. During NF validation, tax information is retrieved from TaxWeb Tax Rules, and NF matching is done with the requisition document. RMS inventory is updated, and then the NF is approved. On NF approval, the shipment is dispatched from the warehouse. RMS publishes the information of the second leg to RWMS through RIB subsequent to the inventory updates. RWMS does not maintain any link between the first and the second legs of the transfer. For RWMS, the two legs of the transfer should be two independent transactions. The second leg does not need to have context type as Repairing. If it is, RWMS ignores it if the recipient is the warehouse.

If the Auto Receive Stock indicator is enabled for the supplier in RMS, the on-hand inventory at the finisher's location is increased immediately as the stock is shipped from the retailer's location. If the indicator is disabled, on shipment of stock from retailer's location, in-transit inventory is increased at the external location. The on-hand inventory is updated only after RMS receives a receipt message from the finisher.

As the shipment leaves the warehouse, the first leg of the transfer is concluded from an ORFM perspective.

**Figure 4–10 RWMS Two-Legged Transfer - First Leg (Shipping)**



## Two-Legged Transfer Receiving at RWMS

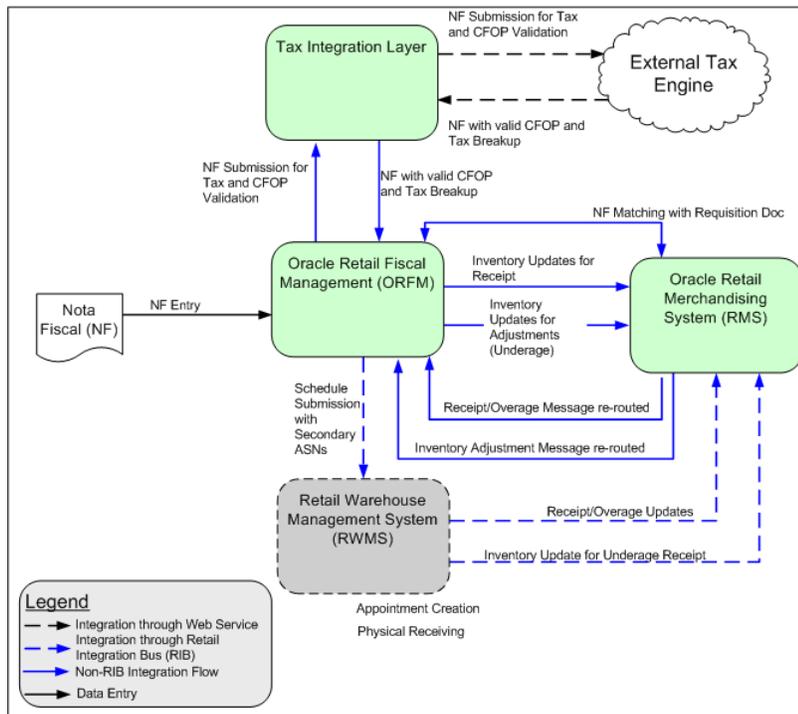
Receiving a two-legged transfer (second leg) is similar to transfer receiving or WO return receiving. The only difference is that the primary ASN is not already available in RWMS when the NF and schedule is entered in ORFM. The finisher ships the finished or repaired items (same or different SKUs) back to the warehouse along with the NF. As the shipment reaches the warehouse, a schedule is created, and the inbound NF is entered into the system. The requisition number for the NF should be the main transfer number (two-legged). Each time, the requisition number on the NF should be the one in RMS. As the NF and schedule go through the tax validation and matching process, the schedule is submitted to RWMS for physical receiving. The message that flows from ORFM to RWMS through RIB should be equivalent to an ASNout message and should contain the schedule number and all the ASN details. The message should be detailed and should contain following fields:

- ASN Number (ORFM should generate the ASN# and maintain the next up sequence number for it).
- Requisition Number (the secondary transfer number)
- Receiving Warehouse Number
- Distinct Item Ids per transfer leg two
- Consolidated Quantities per item-requisition
- UOM
- Schedule number
- Container ID (ORFM should generate the container number and should maintain the next up sequence)

As RWMS receives this message, an automatic appointment (per schedule) is created and the ASN is linked to the appointment. After scheduling the appointment, physical receiving takes place. The user may find discrepancies between the NF and the physical receipt. In case of an overage, RWMS sends a separate overage message. Any over-receipt at RWMS is handled out of the system. In case of under-receipt, the receipt message coming from RWMS contains the same information as the ASN (items and quantities). Any inventory received short should flow to ORFM (through RIB and RMS subscription APIs) as a separate inventory adjustment message.

After ORFM consumes the receipt message, the inventory updates are done in RMS. The in-transit inventory at the destination location is reduced and on-hand inventory is increased. WAC is recalculated for the destination location. Subsequent to the inventory updates for receipts, the inventory should be updated for the inventory adjustment for any under-receipt.

**Figure 4–11 RWMS Two-Legged Transfer - Second Leg (Receiving)**



## Two-Legged Transfer Shipping from SIM

When goods need to be sent from a store to an external finisher for finishing work, a two-legged transfer needs to be initiated in RMS, with the transfer type set as manual requisition and no context type. The first leg of the transfer (store to finisher) should flow to SIM through RIB. SIM should persist this as a return to warehouse (RTW) request.

When goods need to be sent out to external finishers or suppliers for repair, a two-legged transfer can be initiated in RMS, with the transfer type set to manual requisition and the context type to Repairing. The first leg of the transfer (warehouse to finisher) flows to SIM through RIB. SIM persists this as a return to warehouse request.

Because SIM is the physical owner of the inventory, it has better visibility to the goods that need to be repaired. Therefore, a repairing RTW can be initiated in SIM as well. An RTW with the type set to Repairing is created in SIM, it should flow to RMS through RIB. RMS, on consuming the message, creates a two-legged transfer, with the context type set to the value from the transfer created in SIM, and the final location as the same origin store.

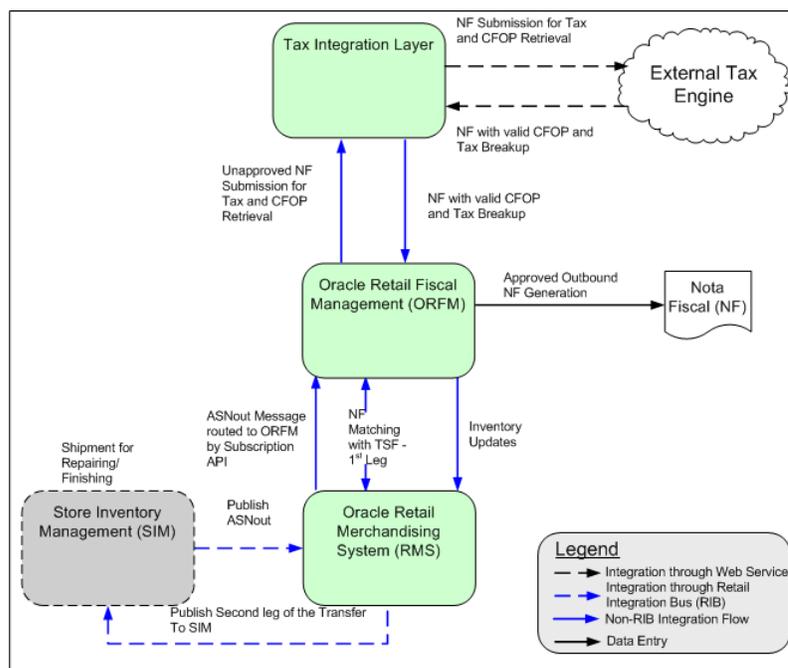
As the required stock is shipped from the store (for the external finisher's location), the ASNout message is published. RMS consumes this message and the subscription API directs it to ORFM. ORFM generates an outbound NF with Repairing as the requisition type. During NF validation, tax information is retrieved from TaxWeb Tax Rules. NF matching is done with the requisition document and on NF approval, the shipment is dispatched from the warehouse. RMS inventory is updated on NF approval. RMS publishes the information of the second leg to SIM through RIB after the inventory updates. SIM needs to maintain a link between the first and the second legs of the transfer because of serial-numbering requirements. RMS, while publishing the second leg to SIM, references the original transfer number. The second leg does not need to

have the context type set to Repairing. SIM persists this transfer as a warehouse delivery.

If the Auto Receive Stock indicator is enabled for the supplier in RMS, the on-hand inventory at the finisher's location is increased immediately as the stock is shipped from the retailer's location. If the indicator is disabled, on shipment of stock from retailer's location, in-transit inventory is increased at the external location. The on-hand inventory is updated only after RMS receives a receipt message from the finisher.

As the shipment leaves the store, the first leg of the transfer is concluded from the ORFM perspective.

**Figure 4–12 SIM Two-Legged Transfer - Shipping**



## Two-Legged Transfer Receiving at SIM

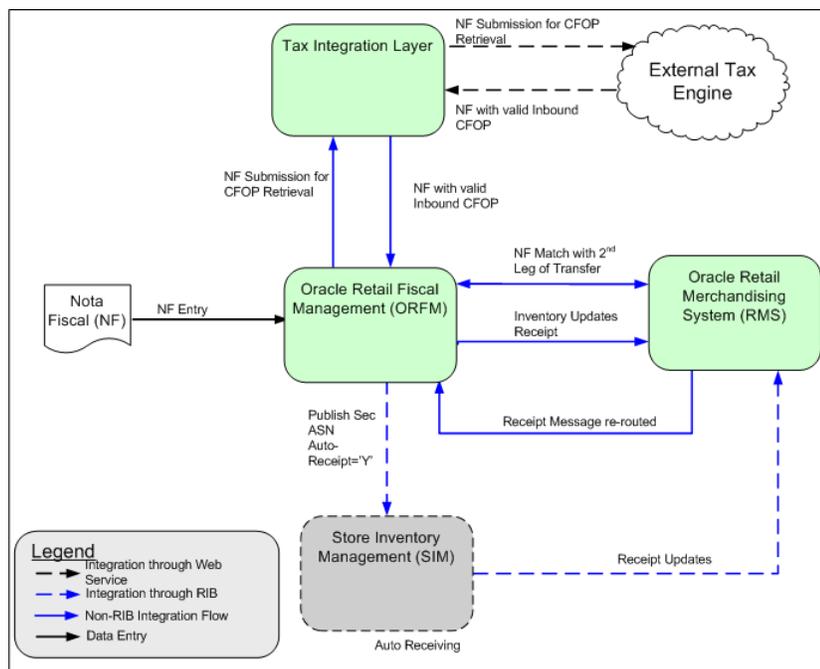
Receiving a two-legged transfer (second leg) is similar to a transfer receiving at a store. The only difference is that the primary ASN is not already available in SIM when the NF and schedule is entered in ORFM. The finisher ships the finished or repaired items (same or different SKUs) back to the warehouse along with the NF. As the shipment reaches the store, a schedule is created, and the inbound NF is entered into the system. The requisition number for the NF should be the main transfer number (two-legged). As the NF and schedule go through the tax validation and matching process, the ASN is published to SIM for physical receiving. The message that flows from ORFM to SIM through RIB should be the equivalent of an ASNout message. The auto-receiving indicator should be 'Y' for this ASN. The detailed message should contain the following information:

- ASN Number (ORFM should generate the ASN number and maintain the next up sequence number for it)
- Requisition Number (the secondary transfer number)
- Receiving store ID

- Distinct Item IDs per transfer leg-two
- Consolidated quantities per item-requisition
- UOM
- Container ID (ORFM should generate the container number and should maintain the next-up sequence)
- Auto-receiving indicator (value should be 'Y')

As SIM receives this message, the ASN is automatically received and a receipt message is sent to ORFM. On receipt of the receipt message from SIM, the inventory updates are done in RMS. The in-transit inventory at the destination location is reduced and on-hand inventory is increased. WAC is recalculated for the destination location.

**Figure 4–13 SIM Two-Legged Transfer - Receive**



## RIB-Based Integration

ORFM can integrate with other Oracle Retail products, such as SIM and RWMS, through Oracle Retail Integration Bus (RIB). RIB utilizes a publish and subscribe (pub/sub) messaging paradigm with some guarantee of delivery for a message. In a pub/sub messaging system, an adapter publishes a message to the integration bus that is then forwarded to one or more subscribers. The publishing adapter does not know or care how many subscribers are waiting for the message, what types of adapters the subscribers are, what the subscribers' current states are (running/down), or where the subscribers are located. Delivering the message to all subscribing adapters is the responsibility of the integration bus.

See the *Oracle Retail Integration Bus Operations Guide* and other RIB documentation for additional information.

## HTTP-Based Integration for TaxWeb

ORFM integrates with RTIL through a HTTP based interfaces. See "[RTIL Architecture](#)" for additional information.

## Nota Fiscal – Receiving and Issuing

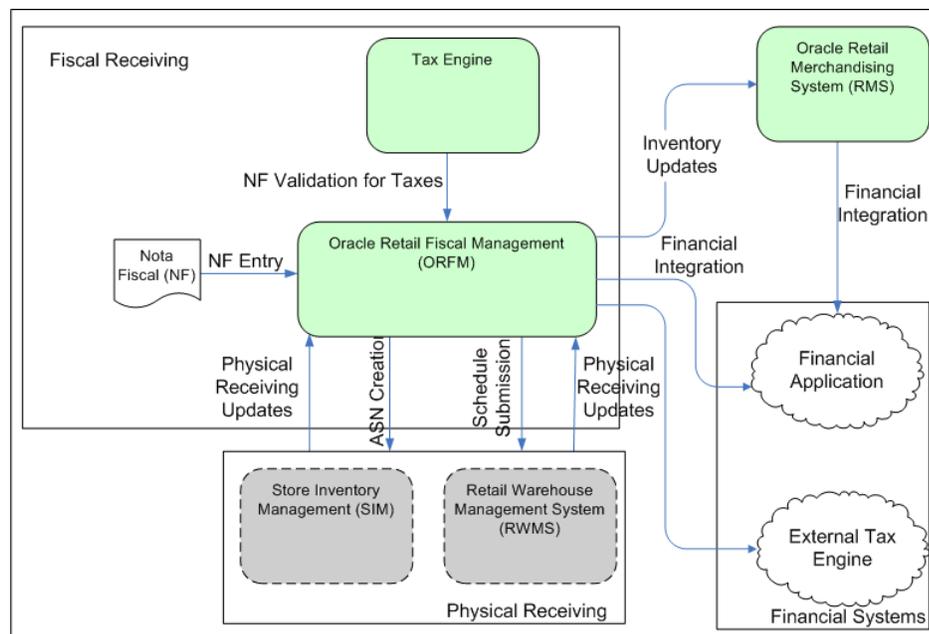
The Nota Fiscal receiving (inbound) and issuing (outbound) are controlled within ORFM and all integration with RMS, RWMS, and SIM is based on the physical movement of the products.

### Nota Fiscal Receiving

When ORFM is enabled, SIM and RWMS are not allowed to physically receive any inventory prior to fiscal receiving (that is, inbound Nota Fiscal processing) is completed in ORFM. After physical receiving in SIM or RWMS, the inventory in RMS is not updated until the fiscal receipt and physical receipt comparison is completed in ORFM and any discrepancy is resolved.

Figure 4-14 outlines the ORFM business process for inbound operations.

**Figure 4–14 Inbound Process Flow**



### Inbound Process Flow

The inbound process flow is as follows:

1. Enter the Nota Fiscal (NF) in the Oracle Retail Fiscal Management (ORFM) system.
2. When the shipment arrives at the warehouse or the store, create a schedule and enter the NFs received.

---

**Note:** You can link more than one NF to a schedule.

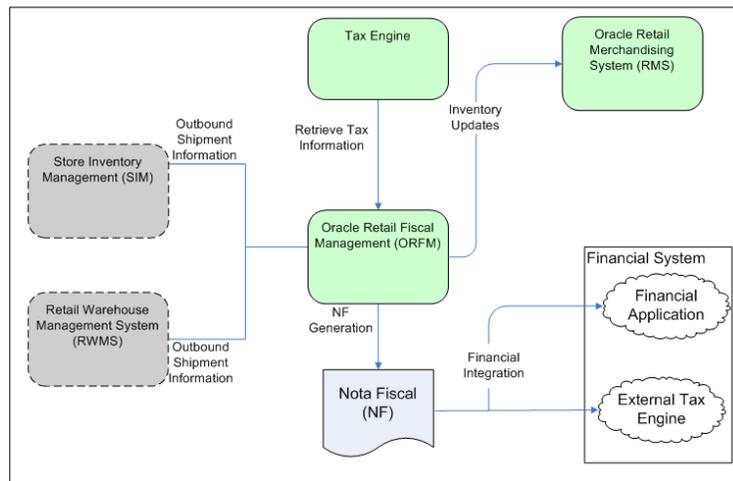
---

3. After NF entry, validate the NFs. In the validation process, the application checks for data integrity. Match the NF with the requisition documents in Retail Merchandising System (RMS). This process is called as Fiscal Receiving.
4. If the NF and the PO does not match, the NF is in discrepancy status. You can identify the following discrepancies with the ORFM application:
  - Unit Cost Discrepancy for each item in the NF
  - Quantity Discrepancy for each item in the NF
  - Tax Discrepancy at aggregate level (such as NF header level or individual item level) for all the applicable taxes
5. After validation, send the schedule to the Warehouse Management System (RWMS), and Store Inventory Management (SIM). After physical receiving, RWMS and SIM publish the receipt updates to ORFM to complete the NF receiving process.
6. After NF processing is complete, ORFM calls RMS to update inventory and WAC.
7. Send the transaction data in RMS and ORFM to a financial application. For fiscal reporting purposes, send the NF data in ORFM to the fiscal reporting system like TaxWeb.

Because the Nota Fiscal contains taxation information, ORFM calls the tax engine for calculations during the receiving and issuing of a Nota Fiscal.

## Nota Fiscal Issuing

When ORFM is enabled, RMS inventory is not updated immediately when any inventory is shipped out from RWMS or SIM. The inventory movement information is first sent to ORFM for the generation of the outbound Nota Fiscal. After approval of the outbound Nota Fiscal, ORFM sends the inventory updates to RMS.



### Outbound Process Flow

The outbound process flow is as follows:

1. Enter the outbound shipment information in SIM or RWMS and send it to ORFM.
2. ORFM sends the outbound shipment information to the tax engine. The tax engine figures the tax information and returns it to ORFM.

3. Create the Nota Fiscal based on the outbound shipment information and the tax information.
4. After NF processing is complete, ORFM calls RMS to update inventory and WAC.

## Triangulation

ORFM supports the triangulation process. Triangulation is a typical purchase transaction where the PO is raised for a supplier (referred to as main/ordered supplier) but the goods are shipped to the retailer by a different supplier (referred to as the delivery supplier). In Brazil, for triangulation transactions, NFs are received from both the main and delivery suppliers. Both the NFs are for the same stock but the details on both the NFs may vary. For example, federal tax such as IPI usually appears on the main supplier's NF, while state taxes such as ICMS and ICMS-ST usually appear on the delivery supplier's NF.

## EDI NF

ORFM supports generation of NF through EDI. Supplier will interface the NF details to RFM and is captured in the EDI tables. RFM has a batch which when run fetches data from the EDI tables and generates the NF in Worksheet status. The NFs are generated with the header, detail, informed header taxes, informed detail taxes, payment information, and complementary details fetched from the EDI tables interfaced by the supplier.

## ORFM and RWMS

RWMS is integrated with the ORFM through RIB for the below business transactions:

### PO Receiving

- ORFM publishes the Appointment Quantity based on the NF details to RWMS
- WMS schedules the Appointment based on the above information
- WMS can over-receive/under-receive during physical receipt. The over-receipt is through new overage screen in RWMS
- The receipt information with the damaged and overage details are published by RWMS to ORFM and ORFM to RMS

### Transfer Receiving

- WMS and ORFM subscribes to the ASN information from the sourcing location
- After the ASN is processed in ORFM, a secondary ASN is published to the warehouse receiving the transfer
- The warehouse will receive against the secondary ASN and the receipt information is interfaced to ORFM through RIB

Brazil requirements are strict for receiving exactly what is on the NF. ORFM provides delivery information to RWMS, from which the user creates receiving appointments. The ability to alter appointment quantities is not available based on system parameter configuration. Validation exists to compare the appointment quantities with the quantities supplied by ORFM. Users can split items as needed by altering case pack sizes, but the totals must match with ORFM. The System Configuration Parameter -- appt\_update\_allowed should be set to N to disallow the appointment quantities being updated during receiving.

Requirements for each appointment are grouped into the same unique schedule number provided by ORFM. That schedule number is added to the appointment table and is searchable by users.

## ORFM and SIM

Nota Fiscal (NF) is a Brazilian fiscal legal document that needs to be generated to follow the physical movement of goods. ORFM handles the specific processing.

SIM is impacted by NF in the following ways:

- Shipping to warehouse or vendor
- DSD Receiving
- Transfer receiving
- Inventory Adjustments and stock counts
- Receive Unit Adjustments

The receiving process in SIM for internal (warehouse and store) deliveries does not allow under receiving or over receiving because of the following:

- Transfers cannot be modified in Brazil through the user interface security; they are auto-received.
- No adjustments are allowed in Brazil.

For warehouse deliveries, NF and SIM obtain the ASN information from the warehouse. After the ASN has been processed in NF, SIM obtains a second ASNin message. This message launches a new auto-receive process. Security is used to limit users' access to the warehouse delivery dialog.

The reason for SIM to still subscribe to the ASNin message from RWMS or another store is so that the system knows which items are on their way while they are in transit, for more accurate inventory information.

## ORFM Integration with Financial Application

Enterprise Business Suite (EBS) is integrated as financial system with Oracle Retail applications including RMS/ReSA. Financial integration between Oracle Retail and EBS is provided in the Oracle Retail Financial Integration (ORFI), Oracle Financial Operations Control Integration Pack for Oracle Retail Merchandising Suite, and E-Business Suite Financials. The Brazilian localized extensions that need to be used in conjunction with the ORFI are discussed in this section.

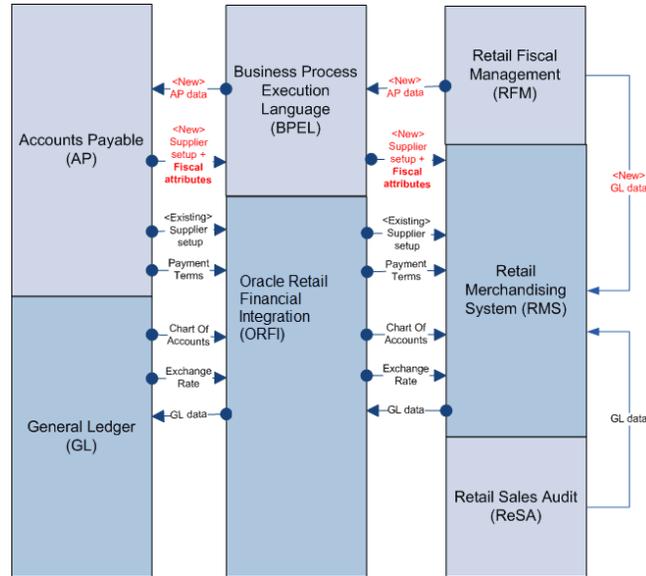
As part of the existing ORFI Integration, the EBS financial module communicates foundation information for supplier setup, payment terms, exchange rates, and chart of accounts to RMS/ReSA. In base version of Oracle Retail, EBS Financial module also communicates Chart of Accounts (COA) to ReIM. In Brazil localization, chart of accounts is not interfaced, and needs to be manually set up in ORFM.

Transaction information is communicated from RMS/ReSA to EBS Financial General Ledger through Oracle Data Integrator (ODI). In base version of Oracle Retail, ReIM communicates payment requests to EBS Accounts Payable (AP) through Retail enabled ODI. In Brazil localization, ORFM has built upon similar interface tables (AP Stage head and detail) as ReIM to publish accounts payable data. These tables are used to publish NF payment data from ORFM to EBS.

The following are discussed as a part of EBS integration:

- Supplier Setup and Integration
- ORFM AP Integration Stage Head and Detail

**Figure 4–15 Flow of information between Financial System and Oracle Retail (Brazil Localization)**



## Supplier Set up and Integration

EBS AP allows setup of Brazilian specific fiscal attributes during supplier setup in a Brazil localized environment, for suppliers located in Brazil. These attributes are passed from EBS to RMS. The Vendor subscription API in RMS is enhanced to subscribe to additional fiscal attributes of the supplier.

These fiscal attributes are passed from EBS AP to RMS in both Supplier Create and Supplier Update/Modify messages flows. Fiscal attributes added to new suppliers/sites that are newly created as well as modifications to these attributes will flow through to RMS as part of the integration.

The global attributes are passed from EBS AP to RMS and consumed at RMS. These attributes are captured in RMS as L10N Attributes (LFAS).

The following table provides the additional attributes that are passed from EBS AP to RMS and consumed at RMS.

EBS Field	Description	RMS Field	Description
global_attribute9	EBS-Inscription Type (CNPJ or CPF)	If type = "CNPJ" then TAXPAYER_TYPE="J" if type = "CPF" then TAXPAYER_TYPE = "F"	Mapped to Taxpayer type field

EBS Field	Description	RMS Field	Description
global_attribute10	EBS-Inscription Number	If type = "CNPJ" concatenate fields global_attribute10, global_attribute11 and global_attribute12 into RMS field CNPJ  If type = "CPF" concatenate fields global_attribute10 and global_attribute12 into RMS field CPF	Concatenate and record value under CNPJ or CPF field
global_attribute11	EBS-Inscription Branch		
global_attribute12	EBS-Inscription Digit		
global_attribute13	EBS-State Inscription	IE	Mapped to IE field
global_attribute14	EBS-City Inscription	IM	Mapped to IM field
global_attribute17	SUFRAMA	SUFRAMA	Mapped to SUFRAMA
NIT	EBS - NIT (Número de Inscrição do Trabalhador perante a previdencia social) - only for individual taxpayers	NIT	Mapped to NIT (Número de Inscrição do Trabalhador perante a previdencia social) - only for individual taxpayers
ISS_CONTRIB_IND	EBS - ISS contributor indicator	ISS_CONTRIB_ID	Mapped to ISS contributor indicator
SIMPLES_IND	EBS - SIMPLES contributor indicator	SIMPLES_CONTRIB_ID	Mapped to SIMPLES contributor indicator
ST_CONTRIB_IND	EBS - ST contributor indicator	ST_CONTRIB_ID	Mapped to ST contributor indicator
RURAL_PROD_IND	EBS - Rural producer	RURAL_PROD_IND	Mapped to Rural producer
ICMS_CONTRIB_IND	EBS - ICMS contributor indicator	ICMS_CONTRIB_IND	Mapped to ICMS contributor indicator
IPI_IND	EBS - IPI contributor indicator	IPI_IND	Mapped to IPI contributor indicator
PIS_CONTRIB_IND	EBS - PIS contributor indicator	PIS_CONTRIB_IND	Mapped to PIS contributor indicator
COFINS_CONTRIB_IND	EBS - COFINS contributor indicator	COFINS_CONTRIB_IND	Mapped to COFINS contributor indicator
IS_INCOME_RANGE_ELIGIBLE	EBS - is income rage eligible indicator	IS_INCOME_RANGE_ELIGIBLE	Mapped to Is Income rage eligible indicator
IS_DISTR_A_MANUFACTURER	EBS - is the distributor a manufacturer indicator	IS_DISTR_A_MANUFACTURER	Mapped to Is the distributor a manufacturer indicator
ICMS_SIMPLES_RATE	EBS - ICMS SIMPLES rate	ICMS_SIMPLES_RATE	Mapped to ICMS SIMPLES rate
Address_line4	EBS - Neighborhood	NEIGHBORHOOD	Mapped to neighborhood

The recommended setting for Brazil localization is that you need to setup 01 address type as externally managed along with 04/06 address types. If you choose to internally manage the 01 address type, then the assumption is that it will be a Brazilian address

and country (same as country on the 06 or 04 address type). RMS will not validate such settings and if you do not follow, it may cause issues in a Brazilian Localized environment.

## ORFM AP Integration Stage Head and Detail

As ReIM is not part of the Brazil solution, the ReIM to AP flow is replaced with an ORFM to AP flow.

Transaction data in ORFM maintains NF information so that it can be sent to the financial application.

Different transaction codes are used for tax-exclusive cost (also referred as BC), cost components (such as freight or insurance), taxes that are input creditable, and taxes that affect cost (non-input credit type).

For tax-related transaction data, the user can configure the system to make a separate entry for each tax code. By doing so, the user can map the same transaction code having different tax codes in different accounts.

Following tables describe the cross mapping between financial staging tables in ORFM and AP:

<b>Product: E-Business Suite</b> <b>Version: 12.1.3</b> <b>Table: AP_INVOICE_LINES_INTERFACE (Target)</b>		<b>Product: Retail ORFM</b> <b>Version: 14.2</b> <b>Table: FM_AP_STAGE_HEAD (Source)</b>		
<b>Column Name</b>	<b>Data Type</b>	<b>Column Name</b>	<b>Data Type</b>	<b>Comments</b>
INVOICE_ID	NUMBER	FISCAL_DOC_ID	NUMBER	Populated with ORFM internal sequencing ID
INVOICE_NUM	VARCHAR2	FISCAL_DOC_NO	VARCHAR2	Populated with the NF fiscal doc ID
INVOICE_TYPE_LOOKUP_CODE	VARCHAR2	FISCAL_TYPE_LOOKUP_CODE	VARCHAR2	Type of Invoice (can be Standard, Credit or Prepayment)
INVOICE_DATE	DATE	TRAN_DATE	DATE	Populated with the NF issue date
VENDOR_ID	NUMBER	VENDOR	NUMBER	Supplier corresponding to the Supplier Site on the NF
VENDOR_SITE_ID	NUMBER	ORACLE_SITE_ID	NUMBER	Supplier site ID of the 'Remit to site' corresponding to the Supplier on the NF
INVOICE_AMOUNT	NUMBER	AMOUNT	NUMBER	
INVOICE_CURRENCY_CODE	VARCHAR2	CURRENCY_CODE	VARCHAR2	Currency based on System Option Setup
EXCHANGE_RATE	NUMBER	EXCHANGE_RATE	NUMBER	

Product: E-Business Suite Version: 12.1.3 Table: AP_INVOICE_LINES_INTERFACE (Target)		Product: Retail ORFM Version: 14.2 Table: FM_AP_STAGE_HEAD (Source)		
Column Name	Data Type	Column Name	Data Type	Comments
EXCHANGE_RATE_TYPE	VARCHAR2	EXCHANGE_RATE_TYPE	VARCHAR2	
TERMS_ID	NUMBER	BEST_TERMS	VARCHAR2	Populated with best terms id picked based on 'Best Terms calculation' logic that uses terms on the Supplier/POs that are present on the NF
SOURCE	VARCHAR2	RFM		Populated with value 'RFM'
ORG_ID	NUMBER	ORG_UNIT	NUMBER	
TERMS_DATE	DATE	BEST_TERMS_DATE	DATE	

Detail/Line Level mappings between financial staging tables in ORFM and AP

Product: E-Business Suite Version: 12.1.3 Table: AP_INVOICE_LINES_INTERFACE (Target)		Product: Retail ORFM Version: 14.2 Table: FM_AP_STAGE_DETAIL (Source)		
Column Name	Data Type	Column Name	Data Type	Comments
INVOICE_ID	NUMBER	FISCAL_DOC_ID	NUMBER	Populated with ORFM internal sequencing ID
LINE_TYPE_LOOKUP_CODE	VARCHAR2	LINE_TYPE_LOOKUP_CODE	VARCHAR2	Type of invoice line (Item, Tax, Miscellaneous)
AMOUNT	NUMBER	AMOUNT	NUMBER	
ACCOUNTING_DATE	DATE	CREATE_DATE_TIME	DATE	
DIST_CODE_CONCATENATED	VARCHAR2	SEGMENT1 - SEGMENT20	VARCHAR2	
ORG_ID	NUMBER	ORG_UNIT	NUMBER	

For information on how to implement EBS, see the following MOS article:

<https://mosemp.us.oracle.com/epmos/faces/DocumentDisplay?id=1580812.1>

---



---

## Publication and Subscription API Designs

This chapter includes publication designs that describe, on a technical level, how ORFM publishes messages to the Oracle Retail Integration Bus (RIB).

The following topics are covered in this chapter:

- [Secondary ASNOut Publication API](#)
- [POSchedule Publication API](#)
- [Shipment Information Subscription API](#)

### Secondary ASNOut Publication API

ORFM publishes this message while submitting a schedule based on PO to SIM for receiving or transfer to RWMS/SIM for receiving. RWMS/SIM subscribes to this message to do the receiving.

This RIB message is triggered by submitting a schedule in ORFM. The output message is in hierarchical structure, with ASN information in the Description section, distro, item, and carton information in the detail sections.

To facilitate the routing of data, the header level of routing information contains the to\_phys\_loc with the value of location ID and 'to\_phys\_loc\_type' with the value W/S and from\_phys\_loc and from\_phys\_loc\_type information. Detail level routing information contains the source\_app with the value of 'ORFM'. This allows RIB to route the message from ORFM to RWMS/SIM.

### Functionality Checklist

Description	ORFM	RIB
ORFM must publish ASNOut information		
Create new publisher	X	X

### Form Impact

None

### Business Object Records

None

## Package Impact

This section covers the impact of the package – FM\_SECONDARY\_ASNOUT.

### Business Object ID

The business object ID for ASNOut publisher is recv\_no.

**Package Name:** FM\_SECONDARY\_ASNOUT

**Spec File Name:** fm\_secondary\_asnout\_sql\_s.pls

**Body File Name:** fm\_secondary\_asnout\_sql\_b.pls

### Package Specification – Global Variables

L\_FAMILY constant varchar2(25) := asnout

### Function Level Description

**ADDTQ** – The inbound Schedule for a transfer is published from ORFM as one ASNOut message to SIM/RWMS and PO published to SIM.

This function adds a record in the FM\_RIB\_RECEIVING\_MFQUEUE table with asnoutmod message\_type for transfer, asnoutcre for PO/Two-legged transfer/Repairing and ASNOut family for each inbound schedule/ASN. The published flag should be set to U.

**GETNXT** – The RIB calls GETNXT function to obtain ASNOut messages. It performs a cursor loop on the unpublished records with PUB\_STATUS = U and FAMILY=asnout on the FM\_RIB\_RECEIVING\_MFQUEUE table. For each record retrieved, GETNXT obtains the following:

- A lock on the queue FM\_RIB\_RECEIVING\_MFQUEUE table for the records which ORFM is going to publish. The lock is obtained by calling the function LOCK\_THE\_BLOCK.
- The information from the FM\_RIB\_RECEIVING\_MFQUEUE table is passed to PROCESS\_QUEUE\_RECORD function. This function will build the Oracle Object message for RIB.
  - If PROCESS\_QUEUE\_RECORD does not run successfully, GETNXT raises an exception call to HANDLE\_ERRORS function that updates the PUB\_STATUS to 'H'.

**PUB\_RETRY** – It performs a cursor loop on the records with PUB\_STATUS = H and FAMILY = asnoutmod. For each record retrieved, PUB\_RETRY obtains the following:

- A lock on the queue FM\_RIB\_RECEIVING\_MFQUEUE table for the records which ORFM is going to publish. The lock is obtained by calling the function LOCK\_THE\_BLOCK.
- The information from the FM\_RIB\_RECEIVING\_MFQUEUE table is passed to PROCESS\_QUEUE\_RECORD function. This function will build the Oracle Object message for RIB.
  - If PROCESS\_QUEUE\_RECORD does not run successfully, PUB\_RETRY raises an exception call to HANDLE\_ERRORS function that updates the PUB\_STATUS to 'H'.

**PROCESS\_QUEUE\_RECORD (local)** – This function is called from GETNXT and PUB\_RETRY functions for those Inbound schedule records that are in PUB\_STATUS =

'U' and 'H' respectively. This function calls `API_LIBRARY.GET_RIB_SETTINGS` to obtain the RIB settings for the ASNOOut message family. It also calls `BUILD_HEADER_OBJECT` function to build the header and detail sections of secondary ASNOOut message. Once the Oracle object is successfully formulated, this function deletes the current record from the queue (that is, `FM_RIB_RECEIVING_MFQUEUE` table) by calling `DELETE_QUEUE_REC` function.

**BUILD\_HEADER\_OBJECT (local)** – This function will take necessary data from `FM_RIB_STG_RECEIVING_HEADER` and `FM_RIB_STG_RECEIVING_DETAIL` tables for the current `schedule_nbr/ASN` and prepare the Oracle object for the ASNOOut message. The structure of ASNOOut is same as base payloads with an additional attribute containing `schedule_nbr` in the `RIB_ASNOOutDesc_REC` section for RWMS.

The same function will build ASNOOut message structure with an additional attribute of `auto_receipt` flag set to Y in the `RIB_ASNOOutDesc_REC` section for SIM. This flag will otherwise be NULL in case of RWMS.

While publishing the secondary ASNOOut to RWMS and SIM, ORFM will be sending three routing information through Oracle Object - `RIB_ROUTINGINFO_REC`.

1. Sending location
2. Receiving location
3. A hard-coded string depicting source application - ORFM

This is required to indicate the source of ASNOOut message is ORFM and hence, it is meant only for RWMS and SIM applications to consume it. However, RMS also subscribes to ASNOOut message, but in this case as the source is ORFM - RMS subscriber package will discard the message.

Call the `BUILD_DETAIL_OBJECTS` to obtain the details of the current schedule record. The `container_qty` is a required field on the RIB object. So, ORFM sends it as 1 instead of NULL.

This function will also be modified to publish the following additional fields in the publishing message to RWMS/SIM:

1. Schedule number (will be NULL for SIM)
2. Auto\_receive flag (will be NULL for RWMS and Y for SIM)

**BUILD\_DETAIL\_OBJECTS (local)** – This function is responsible for building the detail section of ASNOOut message. It builds as many detail Oracle Object as many items present in a transfer. It will fetch the detail records from `FM_RIB_STG_RECEIVING_HEADER`, `FM_RIB_STG_RECEIVING_DETAIL` for the given schedule number and will assign the above details into `RIB_ASNOOutItem_REC`, `RIB_ASNOOutCtn_REC` and `RIB_ASNOOutDistro_REC` record groups.

This package will be modified to generate the ASN# and CTN# and will maintain the next up sequence number for it in case of PO to SIM and two-legged transfer/repairing publishing to RWMS/SIM.

**LOCK\_THE\_BLOCK (local)** – This function locks all queue records for the current business object. This is to ensure that GETNXT does not wait on any business processes that currently have the queue table locked and have not committed.

**HANDLE\_ERRORS (local)** – `HANDLE_ERRORS` is called from GETNXT and `PUB_RETRY` when an exception is raised. If the error is a non-fatal error, GETNXT passes the sequence number of the driving `FM_RIB_RECEIVING_MFQUEUE` record back to the RIB in the `ROUTING_INFO`. It sends back a status of 'H'ospital to the RIB as well. It then updates the status of the queue record to 'H'ospital, so that it will not get picked up again by the driving cursor in GETNXT.

**DELETE\_QUEUE\_REC (local)** – DELETE\_QUEUE\_REC is called from PROCESS\_QUEUE\_RECORD once a queue record is formed from FM\_RIB\_RECEIVING\_MFQUEUE, FM\_RIB\_STG\_RECEIVING\_DETAIL and FM\_RIB\_STG\_RECEIVING\_HEADER tables. This is just to not pick the same record again.

## Trigger Impact

### Trigger Name

Not applicable

### Trigger File Name

Not applicable

### Table

Not applicable

## Message XSD

The file name that corresponds with the message type. Refer the RIB documentation for this message type in order to obtain a detailed picture of the composition of the message.

Message Types	Message Type Description	XML Schema Definition (XML)
ASNOutCre	ASNOut Create message	ASNOutDesc.xsd
ASNOutMod	ASNOut Modify message	ASNOutDesc.xsd

## Table Impact

Table	Select	Insert	Update	Delete
FM_RIB_RECEIVING_MFQUEUE	Y	Y	Y	Y
FM_RIB_STG_RECEIVING_HEADER	Y	Y	N	Y
FM_RIB_STG_RECEIVING_DETAIL	Y	Y	N	Y
FM_RECEIVING_HEADER	Y	N	N	N
FM_RECEIVING_DETAIL	Y	N	N	N

## Design Assumptions

This message is applicable only if ORFM exists in case of Brazil countries. It is applicable for PO/Schedule publication to SIM and transfer/schedule publishing to RWMS/SIM.

## POSchedule Publication API

ORFM publishes this message while submitting a schedule based on 'PO' to RWMS for receiving. RWMS subscribes to this message in order to create schedule based appointment and does the receiving.

This RIB message is triggered by submitting of PO based Schedule in ORFM. The output message is in hierarchical structure, with Schedule no in the Description section, PO information in the Header and Item details in the detail section.

To facilitate the routing of data, the header level of routing info contains the to\_phys\_loc with the value of location ID and to\_phys\_loc\_type with the value W. Detail level routing info contains the source\_app with the value of ORFM. This allows the RIB to route the message from ORFM to RWMS.

### Functionality Checklist

Description	ORFM	RIB
ORFM must publish POSchedule information		
Create new publisher	X	X

### Form Impact

None

### Business Object Records

None

### Package Impact

This section covers the impact of the package – FM\_SCHED\_SUBMIT.

#### Business Object ID

The business object ID for POSchedule publisher is recv\_no that is, schedule no.

**Package Name: FM\_SCHED\_SUBMIT**

**Spec File Name: fm\_sched\_submit\_sql\_s.pls**

**Body File Name: fm\_sched\_submit\_sql\_b.pls**

#### Package Specification - Global Variables

FAMILY constant varchar2(25) := poschedule

LP\_cre\_type varchar2(15) := poschedulecre.

#### Function Level Description

**ADDTOQ** – This function adds the PO Schedule to the FM\_RIB\_RECEIVING\_MFQUEUE table in 'U'npublished status. It stages the recv\_no (schedule number) for publishing to the RIB.

**GETNXT** – This procedure is called from the RIB to obtain the next 'U'npublished schedule number from FM\_RIB\_RECEIVING\_MFQUEUE table for publishing. Based

on the seq\_no on FM\_RIB\_RECEIVING\_MFQUEUE, it calls the PROCESS\_QUEUE\_RECORD procedure.

GETNXT obtains the following:

- A lock on the queue FM\_RIB\_RECEIVING\_MFQUEUE table for the records which ORFM is going to publish. The lock is obtained by calling the function LOCK\_THE\_BLOCK.
- The information from the FM\_RIB\_RECEIVING\_MFQUEUE table is passed to PROCESS\_QUEUE\_RECORD function. This function will build the Oracle Object message for RIB.

If PROCESS\_QUEUE\_RECORD does not run successfully, GETNXT raises an exception.

- If any exception is raised in GETNXT, including the exception raised by an unsuccessful call to PROCESS\_QUEUE\_RECORD, HANDLE\_ERRORS is called.

**PUB\_RETRY** – It will process the record for PUB\_STATUS = 'H' and MESSAGE\_TYPE = poschedcre. For each record retrieved, PUB\_RETRY obtains the following:

- A lock on the queue FM\_RIB\_RECEIVING\_MFQUEUE table for the records which ORFM is going to publish. The lock is obtained by calling the function LOCK\_THE\_BLOCK.
- The information from the FM\_RIB\_RECEIVING\_MFQUEUE table is passed to PROCESS\_QUEUE\_RECORD function. This function will build the Oracle Object message for RIB.
- If PROCESS\_QUEUE\_RECORD does not run successfully, PUB\_RETRY raises an exception. Call to HANDLE\_ERRORS function that updates the PUB\_STATUS to 'H'.
- If any exception is raised in GETNXT, including the exception raised by an unsuccessful call to PROCESS\_QUEUE\_RECORD, HANDLE\_ERRORS is called.

**PROCESS\_QUEUE\_RECORD (local)** – This function is called from GETNXT and PUB\_RETRY functions for those Inbound schedule records that are in PUB\_STATUS = U and H respectively. This function calls API\_LIBRARY.GET\_RIB\_SETTINGS to obtain the RIB settings for its family (posched). It also calls BUILD\_HEADER\_OBJECT function to build header and detail sections of the PO Schedule Oracle objects. Once the Oracle object is successfully formulated, this function deletes the current record from the queue (that is, FM\_RIB\_RECEIVING\_MFQUEUE table) by calling DELETE\_QUEUE\_REC function.

**BUILD\_HEADER\_OBJECT (local)** – Take all necessary data from FM\_RIB\_STG\_RECEIVING\_HEADER and FM\_RIB\_STG\_RECEIVING\_DETAIL tables for the current schedule and put it into a RIB\_POScheduleDesc\_REC object. Two routing information (source Application that is, ORFM and location that is, Warehouse) has to be sent to RIB through RIB\_ROUTINGINFO\_REC for PO.

This function publishes the following fields in the header section of the POSchedule message to RWMS:

- schedule\_nbr
- receiving\_location\_id

This function will give a call to the BUILD\_DETAIL\_OBJECTS function.

**BUILD\_DETAIL\_OBJECTS (local)** – The function is responsible for building detail level Oracle Objects. It fetches the detail records from FM\_RIB\_STG\_RECEIVING\_

DETAIL for the given schedule number and formulates the message structures: RIB\_POSchedule\_REC and RIB\_POScheduleDtl\_REC.

This function publishes the following fields to the above Oracle objects:

- requisition\_nbr
- requisition\_type
- item\_id
- consolidate\_qty

**LOCK\_THE\_BLOCK (local)** – This function locks all queue records for the current schedule\_nbr. This is to ensure that GETNXT does not wait on any business processes that currently have the queue table locked and have not committed.

**HANDLE\_ERRORS (local)** – HANDLE\_ERRORS is called from GETNXT and PUB\_RETRY when an exception is raised. If the error is a non-fatal error, GETNXT passes the sequence number of the driving FM\_RIB\_RECEIVING\_MFQUEUE record back to the RIB in the ROUTING\_INFO. It sends back a status of 'H'ospital to the RIB as well. It then updates the status of the queue record to 'H'ospital, so that it will not get picked up again by the driving cursor in GETNXT.

If the error is a fatal error, a status of 'E'rror is returned to the RIB.

**DELETE\_QUEUE\_REC (local)** – DELETE\_QUEUE\_REC is called from PROCESS\_QUEUE\_RECORD once a queue record is formed from FM\_RIB\_RECEIVING\_MFQUEUE, FM\_RIB\_STG\_RECEIVING\_DETAIL, FM\_RIB\_STG\_RECEIVING\_HEADER tables. This is just to not pick the same record again.

## Trigger Impact

### Trigger Name

Not applicable

### Trigger File Name

Not applicable

### Table

Not applicable

## Message XSD

The file name that corresponds with the message type. Refer to the RIB documentation for this message type in order to obtain a detailed picture of the composition of the message.

Message Types	Message Type Description	XML Schema Definition (XSD)
POScheduleCre	POSchedule Create message	POScheduleDesc.xsd

## Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
FM_RIB_RECEIVING_MFQUEUE	Y	Y	Y	Y
FM_RIB_STG_RECEIVING_HEADER	Y	Y	N	Y
FM_RIB_STG_RECEIVING_DETAIL	Y	Y	N	Y
FM_RECEIVING_HEADER	Y	N	N	N
FM_RECEIVING_DETAIL	Y	N	N	N

## Design Assumptions

This message is applicable only if ORFM exists for Brazil localization. It is applicable for PO/Schedule publication to RWMS only.

## Shipment Information Subscription API

The business overview, functionality checklist, and package impact of the Shipment Information Subscription API is covered in the subsequent sections.

## Business Overview

ORFM subscribes this message in order to be able to generate and approve the NF-e.

This RIB message is triggered by RWMS and SIM for the following scenarios:

- Customer Order Fulfillment (direct customer delivery)
- Customer Order Fulfillment (multi-site fulfillment)
- Stock Orders (regular transfers between locations)
- RTVs\*

## Assumptions

RWMS does not support NF Request for RTVs. Only SIM can publish ShipInfo message for RTV transactions. Because in SIM the RTV and RTW are handled in the same context. ORFM will be able to consume ShipInfo for both, however only for RTW the message will trigger NF creation. For RTV the ShipInfo message will be ignored and NF will be created based on physical shipment and ASNOut message.

To facilitate the routing of data, RFM must be able to identify the transaction type that is being performed for each NF Request message. Some fields in the message structure will be leveraged to address this need.

The following table contains all valid combinations and the fields to be used to differentiate the transactions:

distro_doc_type	Consumer_direct	to_location	Transaction Type for RFM
T	Y	Non-stockholding	Direct delivery CO transfer (NF request created at WH) Flow link

<b>distro_doc_type</b>	<b>Consumer_direct</b>	<b>to_location</b>	<b>Transaction Type for RFM</b>
C	Y	Null	*Customer pick up at store (NF request created at store) Flow link
T	Y	Stockholding	Multi-site CO transfer Flow Link WH ST
T	N	Stockholding	Regular location to location transfer Flow Link WH ST
S	N	Stockholding	Regular location to location transfer Flow Link WH ST
A	N	Stockholding	Regular location to location transfer Flow Link WH ST
Null for RTV and 'T' for RTW	N	Vendor ID or WH ID	Returns from store to Vendor or WH Flow Link ST

Invalid combinations will result in error while consuming the ShipInfoDesc message which will be rejected in ORFM.

## Functionality Checklist

None

## Form Impact

None

## Business Object Records

None

## Package Impact

This section covers the impact of the package – FM\_SHIPINFO\_SQL.

**Package Name:** FM\_SHIPINFO\_SQL

**Spec File Name:** fm\_shipinfo\_sql\_s.pls

**Body File Name:** fm\_shipinfo\_sql\_b.pls

### Package Specification - Global Variables

LP\_shipinfo\_type varchar2(15) := 'shipinfocre'

### Function/Procedure Level Description

**CONSUME** – The RIB calls CONSUME procedure in order to be able to generate and approve the NF-e.

This procedure call function CONSUME\_SHIPINFO responsible to make a bridge of process expected.

The function will be calling FM\_STG\_SHIPINFO\_SQL.CONSUME add records into FM\_STG\_ASNOUT\_DESC, FM\_STG\_ASNOUT\_DISTRO, FM\_STG\_ASNOUT\_CTN and FM\_STG\_ASNOUT\_ITEM, validating message received based in the combination of values mentioned previously.

When everything is performed, CONSUME\_SHIPINFO will be calling FM\_EXIT\_NF\_CREATION\_SQL to generate the NF-e.

## Trigger Impact

### Trigger Name

Not applicable

### Trigger File Name

Not applicable

### Table

Not applicable

## Message XSD

The file name that corresponds with the message type. Refer the RIB documentation for this message type in order to obtain a detailed picture of the composition of the message.

Message Types	Message Type Description	XML Schema Definition (XSD)
ShipInfoCre	ShipInfo Create message	ShipInfoDesc.xsd

## Table Impact

Table	Select	Insert	Update	Delete
FM_STG_ASNOUT_DISTRO	Y	Y	Y	Y
FM_STG_ASNOUT_DESC	Y	Y	Y	Y
FM_STG_ASNOUT_CTN	Y	Y	Y	Y
FM_STG_ASNOUT_ITEM	Y	Y	Y	Y

## Design Assumptions

This message is applicable only if ORFM exists for Brazil localization. It is applicable to generate and approve the NF-e subscribed by RWMS/SIM only.

---

---

## Archiving and Purging Data

In a production environment, the number of transactions increases over a period of time. To keep performance acceptable, data must be purged periodically from the active tables of the application. ORFM purges the data from the active business tables and stores the data in history tables.

To enter a Nota Fiscal (for a PO, TO, or RTV), create a schedule in ORFM. After the schedule is in Financial Posted status and passes a certain number of days, all related data with that schedule are purged from the transaction tables. In addition, all the NFs that are inactive for a schedule are purged. An inactive schedule is also purged from the transaction table and the data is stored in the history table.

For more information about purge batches, see [Chapter 11, "Maintaining Tax Related Data - TaxWeb"](#).



---

---

## Integrating with TaxWeb Tax Rules– Retail Tax Integration Layer

Brazil has complex tax and fiscal systems. In order to have a flexible format to fit any kind of tax scenario and also to make the tax rules setup part of a specialist solution, Oracle Retail has integrated its applications with third-party tax engines provided by Oracle partners. The ORFM system is integrated with the TaxWeb Tax Rules solution through the Oracle Retail Tax Integration Layer (RTIL).

RTIL acts as a connector that exposes a Retail Tax Data Model API for calculating tax, which is envisioned to be an abstraction over the TaxWeb Tax Rules tax data model. RTIL hosts the TaxWeb Tax Rules Adapter that is responsible for converting the Retail tax data model into the native language of a tax service provider and communicating with a third-party tax solution service. RTIL supports integration of RMS and ORFM with the TaxWeb Tax Rules. The ORFM system is integrated with the TaxWeb Tax Rules solution to obtain information about all Brazilian tax legislation, with a high level of exception treatments.

For all flows in Oracle Retail that need to have tax calculation, the integration layer is used to have all taxes applied from the TaxWeb Tax Rules, considering the input parameters.

The following processes in ORFM/RMS use the tax calculation integration:

- Inbound NF validation
- Outbound NF issuance
- PO tax breakdown
- Item creation
- Refresh tax
- Item fiscal reclassification

For each of these processes, the needed input information is sent to the TaxWeb Tax Rules through RTIL. RTIL is responsible for transforming the request information from the Retail tax data model format to the TaxWeb Tax Rules specific format, and vice versa for the response information.

This chapter covers the following topics:

- [Integration Overview](#)
- [Retail Tax Data Model](#)
- [RTIL Architecture](#)
- [RTIL Configuration](#)

- [Input Parameters Data Mapping and Expected Output Results](#)
- [Concurrent Processing in RTIL](#)
- [Tax Calculation Scenarios](#)

## Integration Overview

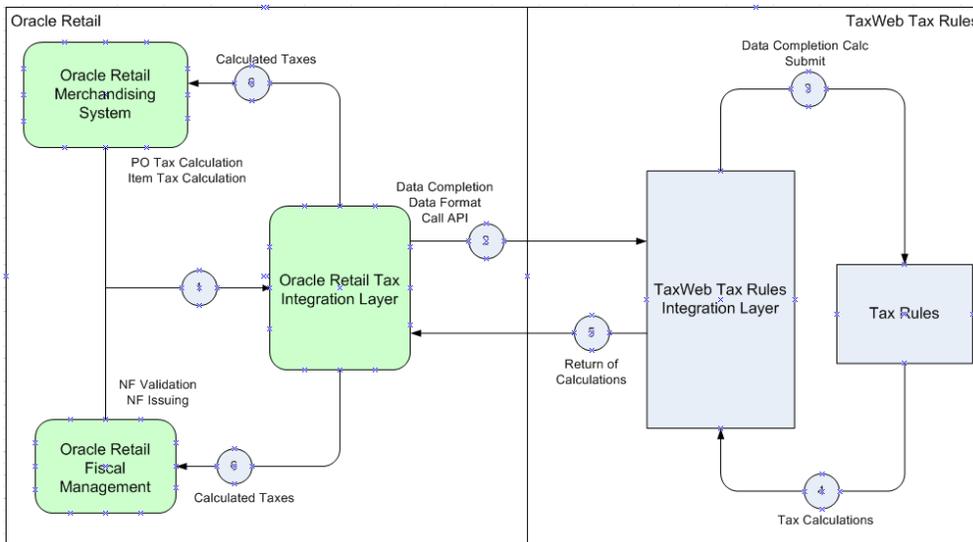
RTIL is responsible for the TaxWeb Tax Rules API call with the necessary formatting. The TaxWeb Tax Rules API exposed has a native Nota Fiscal format, and formatting must be sent in that way for tax calculation in TaxWeb Tax Rules.

In TaxWeb Tax Rules, an integration layer is used for data completion. Some of the attributes in the API must be determined by Tax Rules (such as CFOP). Only after this step is complete the tax classification is triggered internally in the TaxWeb Tax Rules. Once the taxes are classified, the API with the returned values is sent back to the integration layer in Oracle Retail.

The layout of the TaxWeb Tax Rules solution is based on the NF layout and has the following structure:

- Entities (foundation data and fiscal attributes of each entity of a NF, such as Issuer, Addressee, and Transporter)
- NF header information
- NF item information
- Tax classification (enquadramento)

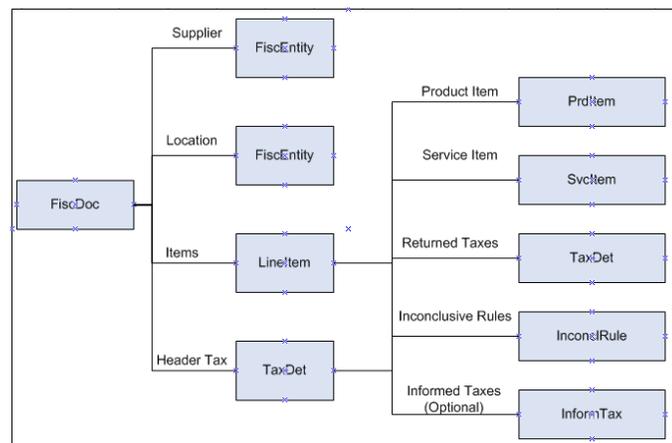
**Figure 7-1 Oracle Retail/TaxWeb Tax Rules Integration**



## Retail Tax Data Model

The Retail Tax Data model is an abstraction introduced over the TaxWeb Tax Rules Tax Data model with the aim of achieving semantic interoperability. The Retail Tax Data Model aims to decouple Oracle Retail Products (RMS, ORFM) from the vendor specific data model. This model enables RMS/ORFM to deal with a more generic format than being tightly locked with TaxWeb Tax Rules Vendor data model. This model allows applications to be more loosely coupled.

Figure 7-2 Retail Data Model Structure



## Object Structure Overview

- **FiscDoc** – This object represents a fiscal document and eventually maps to the Nota Fiscal of the TaxWeb Tax Rules in the integration with the TaxWeb Tax Rules. It has the information related to the NF header and it has included the other objects that detail each part of the fiscal document as the issuer, addressee, and items.
- **FiscEntity** – This object has the fiscal and master information related to the entities in a fiscal document. This object will be used to detail the issuer/supplier, addressee/location and facilitator/transporter information.
- **LineItem** – This object represents the detail of a line item in a fiscal document. It has included the taxes at item level.
- **TaxDet** – This object represents the details of an individual tax. It can be attached at the Fiscal Document level or at a DocLineItem Level. Each tax applied to an item will have a different object of this type.
- **InformTax** – This object represents the details of an individual tax informed by the calling application and supplied in the request. This indicates that the TaxWeb Tax Rules should not compute this specific tax but can use this information to derive other taxes if they are dependent on the informed tax. This is an optional entity which is sent in scenarios where in the NF tax is assumed to be the correct one. It can be attached at the Fiscal Document level or at a LineItem level.
- **InconclRule** – This object has the rules and laws applied to each tax. It has the regulation and the log of the tax classification. It notifies the rules that are not applied to the item and classified as inconclusive rules. This can be used as an information log to the user to show potential tax setup problems.
- **ProductItem/ServiceItem** – Code and description of items which can be products or services. These objects will have the fiscal attribute details for the items depending on the item type (product or service).
- **FiscDocChnk** – This is a placeholder object introduced to support a logical unit of work (contains one or more fiscal documents) that can be sent in a single request to the TaxWeb Tax Rules. It does not have a direct counterpart in the TaxWeb Tax Rules data model but aggregates to a collection of DocFiscal objects in the vendor data model. A collection of FiscDocChnk objects helps in enabling RTIL to support splitting a large request into multiple chunks and process them concurrently to

enhance performance and throughput. Aggregation of the responses from the TaxWeb Tax Rules are handled internally in RTIL.

## Retail Tax Data Model to TaxWeb Tax Rules Data Model Object Mapping

The following table illustrates the mapping of the Retail Tax Data Model to TaxWeb Tax Rules at an object level.

Retail Tax Object	TaxWeb Tax Rules Object
FiscDoc	DocFiscal
FiscEntity	Pessoa
LineItem	ItemDocFiscal
TaxDet	Enquadramento
InconclRule	EnquadramentoItem
ProductItem	Produto
InformTax	Enquadramento
ServiceItem	Servico

## RTIL Architecture

RTIL exposes a servlet controller to receive the HTTP request from ORFM/RMS application in a synchronous request/response pattern. Based on the http request from ORFM/RMS, RTIL will fetch the data from RMS/ORFM and form the request for external tax vendor. RTIL will invoke the associated external third-party services with native POJO call to obtain the tax details which will be passed back to the calling application.

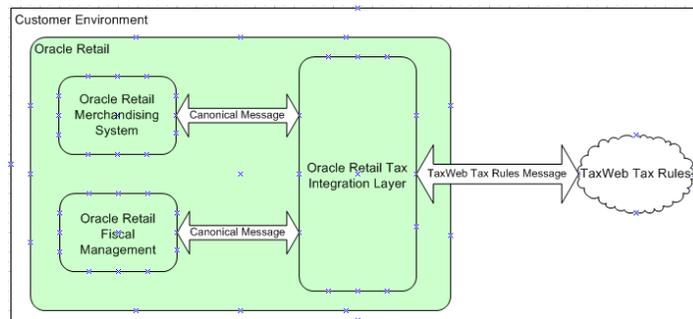
Retail Tax Integration Layer will be implemented as Java EE application with the associated tax service provider adapters. This layer will form the conduit between the Oracle Retail Applications and the Tax service provider. The Retail Tax Integration layer is responsible for assembling and disassembling of the vendor specific data model to Retail tax data model based on the configuration. The Retail Tax Integration layer is envisioned to host vendor specific connectors which can communicate to the external third party services. The subscribing application is not aware of the tax service provider, RTIL acts as a bridge between the subscribing application and third-party tax service provider. In the current release, RTIL will host a TaxWeb Tax Rules adapter and interacts with the TaxWeb Tax Rules using a POJO interface.

## High Level Integration View

The integration layer in Oracle Retail does the following.

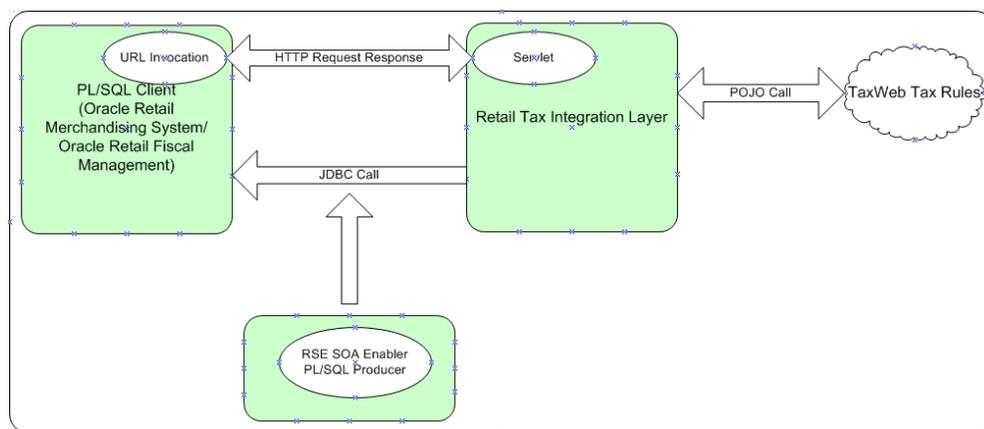
- Exposes a more abstracted tax data format (Retail Tax Data Model) for RMS/ORFM to pass their transaction details that require taxes to be computed
- Performs the data transformation from canonical to the TaxWeb Tax Rules specific format
- Invokes the taxation API of the TaxWeb Tax Rules
- Transforms the response containing taxes from the TaxWeb Tax Rules specific format back to the canonical format to be consumed by RMS/ORFM
- RTIL is used as the unique tax calculation entry point

**Figure 7-3 RTIL High Level Integration**



## RTIL Integration Architecture

**Figure 7-4 RTIL Integration Architecture**

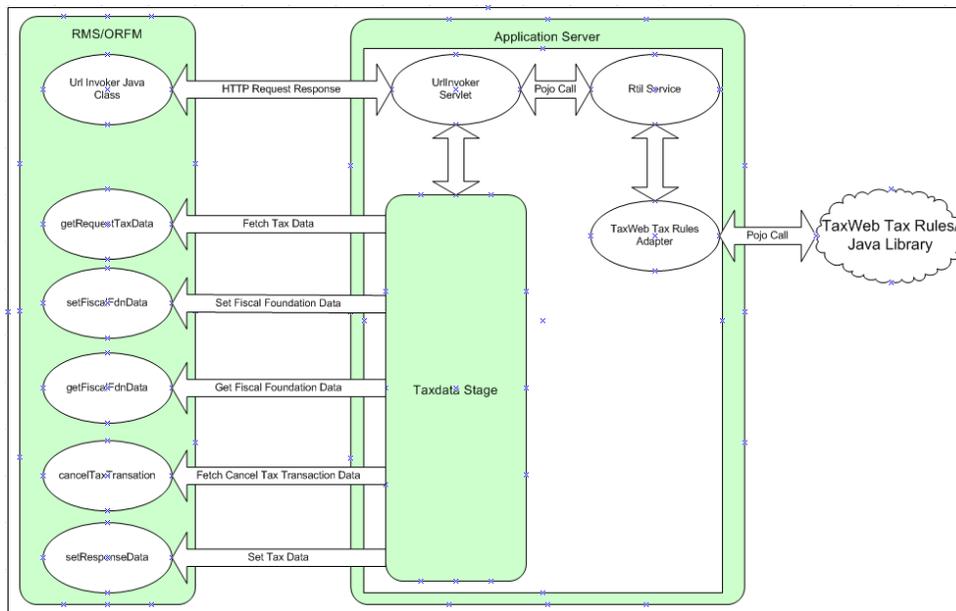


The above diagram illustrates the high-level components and its interactions along with the run time and the design time view of the system. The Retail Service Enabler Utility is used to generate the necessary PL/SQL producer and its associated artifacts to fetch/set tax staged data from the database.

The client uses an URL Invoker class signaling RTIL to fetch the tax request data from the client's staged tables and the TaxWeb Tax Rules API will be invoked as a POJO (Plain Old Java Object) call from RTIL.

## Components Deployment View

**Figure 7-5 Components Deployment View**



The preceding diagram illustrates the deployment view of the components. The salient features of the architecture are:

- Retail applications interact only with tax services in an abstracted tax data format using RTIL-supplied client components.
- RTIL insulates the Oracle Retail applications from tax service providers. RTIL will be deployed as an enterprise application instance (EAR file) in Oracle WebLogic Application Server.
- RTIL will host the vendor adapters.
- The adapters interact with the tax service providers in native format.
- The adapters are responsible for converting the Retail Tax Data Model to the application-specific data model and vice versa.
- Tax service providers can be deployed on the same or a different application server, based on customer needs.
- The flow typically involves RTIL being invoked by a HTTP request which contains the qualifying information to invoke the appropriate the TaxWeb Tax Rules API.
- The primary group of operations supported are:
  - Fetch Tax Staged Data from RMS/ORFM and invoke the TaxWeb Tax Rules calculation API through a POJO call
  - Update the Tax Response on to the Staged Tables in RMS/ORFM
  - Support invocation of the TaxWeb Tax Rules tax fiscal foundation data and updation of foundation response back to RMS/ORFM
  - Invocation of a cancel Tax Transaction API on the TaxWeb Tax Rules (This is a compensation API to be invoked to eliminate stale data in the TaxWeb Tax

Rules in case of failures of retail transactions in RMS/ORFM post the TaxWeb Tax Rules data updates).

## Client Components

**URL Invoker** – This is a java class loaded into the database JVM, which triggers RTIL to fetch the Tax request data from the RMS/ORFM staged tables. It's a synchronous call, which is blocked till RTIL updates the tax response from the TaxWeb Tax Rules back on to the RMS/ORFM staged tables.

It will be the responsibility of the URL Invoker to send a unique key to RTIL, which will be associated with the tax request and tax response that RTIL has to fetch and set. Also, the differentiator parameters for the tax call or to fetch different foundation data or to invoke tax compensation API has to be sent by the invoker to trigger appropriate processing in RTIL.

**Retail Tax API** – This is a TaxAPI which is in a more abstracted format that is exposed by RTIL to RMS/RFM. Any data enrichment if required should be done prior to the invocation of this API Call. This API will be in the format of the Oracle Retail application implementation.

Since RMS/ORFM are Forms/PLSQL applications, the Oracle Object Types representing abstracted Tax Request and Response are generated from Functional Artifact Generator tool for the given RTIL Tax XSD's. This ensures the API is in the implementation format of the Oracle Retail application.

Based on the request parameters from the URL Invoker, appropriate tax API will be called from RTIL.

**TaxData Stage component, PL/SQL packages** –This is generated by RSE and provides the runtime support for fetching/setting tax data from/to the database staged tables.

## RTIL Configuration

RTIL broadly supports the following configuration features:

- Logging/auditing
- User-friendly exception messages

## Logging/Audit

Logging is implemented in a declarative manner using Spring Aspect Oriented Programming (AOP). AOP is a paradigm that addresses separation of concerns, thereby ensuring that logging and business logic are kept separate in the code, avoiding code cluttering. Logging is introduced during run time in a noninvasive manner leveraging spring AOP features. The logging utilities are implemented using the industry-standard Apache commons logging API.

The following are the classes that act as the logging aspect to the application and apply to different methods of the application as configured.

- LoggerAspect
- ObjectLoggerAspect
- TimerLoggerAspect

## LoggerAspect

This class logs entry and exit statements for the methods in core RTIL classes. This also logs if there is some exception during the execution of these methods.

## ObjectLoggerAspect

This class logs the entire content of request and response objects used in the transactions. It also logs the request object content and response object contents both in the canonical and native TaxWeb Tax Rules format. This is useful for troubleshooting the request and responses.

## TimerLoggerAspect

This logs the time taken in milliseconds for execution of the methods present in assembler, Gateway, and adapter classes. This aspect can be used to obtain a timing profile of the method executions.

These aspects are declared by annotations and registered in applicationContext.xml.

The following is a snapshot of the applicationContext.xml configuration file which shows the configuration for registering Aspects.

```
<aop:aspectj-autoproxy />
<bean id="timingLoggerAspect"
class="com.oracle.retail.tax.aspects.TimingLoggerAspect" />
<bean id="loggerAspect" class="com.oracle.retail.tax.aspects.LoggerAspect" />
<bean id="objectLoggerAspect"
class="com.oracle.retail.tax.aspects.ObjectLoggerAspect" />
```

---

**Note:** Removal of these entries in this configuration file is equivalent to removing the code which logs these details. Thus, the enablement/disablement of logging is controlled in a declarative manner using Spring AOP.

---

A second level of control is available in the log4j.properties file. This file controls the following logging features whose configuration can be changed to suit customer needs:

- Logging at different granularities (INFO, ERROR, DEBUG, FATAL)
- File size limit for each log file (log4j.appender.LOGFILE.MaxFileSize)
- Number of log files to be retained during logfile rollover (log4j.appender.LOGFILE.MaxBackupIndex)

The following is a snapshot of the log4j.properties configuration file supplied with default options during deployment time.

```
# Root Level Logger
log4j.rootLogger=INFO,STDOUT

log4j.appender.STDOUT=org.apache.log4j.ConsoleAppender
log4j.appender.STDOUT.layout=org.apache.log4j.PatternLayout
log4j.appender.STDOUT.layout.ConversionPattern=%d %p [%c] - %m%n

log4j.appender.LOGFILE=org.apache.log4j.RollingFileAppender
log4j.appender.LOGFILE.MaxFileSize=5MB
log4j.appender.LOGFILE.File=./log/rtil.log
# Keep ten backup files.
log4j.appender.LOGFILE.MaxBackupIndex=30
```

```
# Pattern to output: date priority [category] - message
log4j.appender.LOGFILE.layout=org.apache.log4j.PatternLayout
log4j.appender.LOGFILE.layout.ConversionPattern=%d %p [%c] - %m%n

#Class Level logger
log4j.category.org.dozer.MappingProcessor=INFO,STDOUT,appender
log4j.category.com.oracle.retail.tax.aspects.TimingLoggerAspect=DEBUG,LOGFILE
log4j.category.com.oracle.retail.tax.aspects.ObjectLoggerAspect=DEBUG,LOGFILE
log4j.category.com.oracle.retail.tax.aspects.LoggerAspect=DEBUG,LOGFILE

log4j.rootCategory= ALL,LOGFILE,STDOUT
```

### User-friendly Exception Messages

RTIL provides some flexibility in configuring user-friendly error messages based on the exception conditions encountered. The `ExceptionMessage.properties` file contains the list of system-generated exception messages in RTIL, which can be further mapped to the user messages required by the customer. The file is a typical properties file containing key value pairs. This file is read by RTIL infrastructure to substitute the system generated message pattern (Key) with the supplied user message (value), and the user message is propagated to the calling infrastructure (ORFM/RMS).

---

---

**Note:** The configuration option provided is only for supplying/altering the user defined messages. This file does not support the addition/modification of system defined messages.

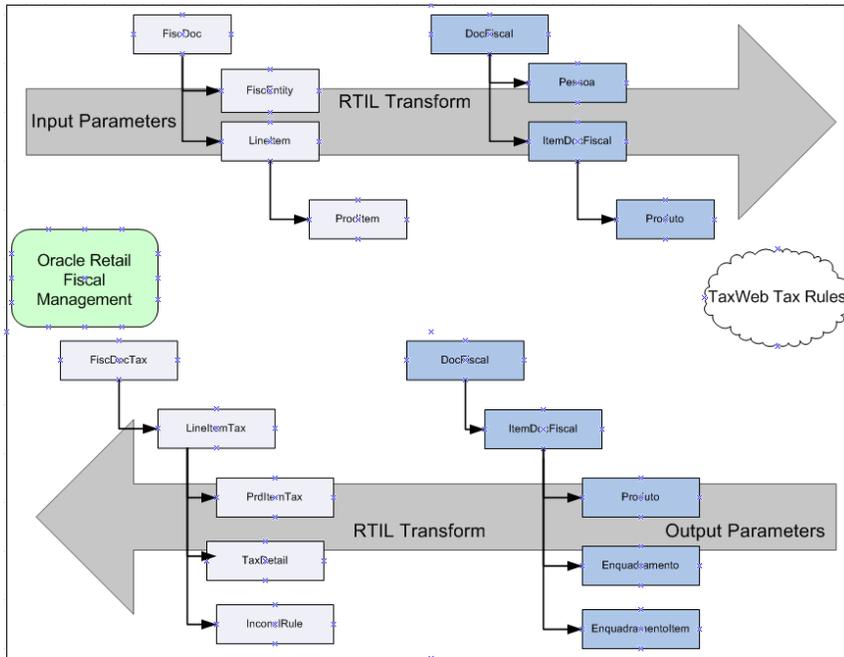
---

---

## Input Parameters Data Mapping and Expected Output Results

ORFM calls the Tax Rules API for the expected scenario and considering the structure of the API, which is similar to a fiscal document. All the input parameters are formatted in that structure, and the returned values are also sent in the same format. The input parameters do not contain any tax information. Only the fiscal attributes and values are sent. In the output parameters, the tax calculation results are sent back based on the original values in the same fiscal document format.

**Figure 7-6 Input x Output Parameters**



## Concurrent Processing in RTIL

RTIL being a J2EE application and deployed in WebLogic application server inherently supports parallel processing of requests, provided its clients (RMS/ORFM) can send the requests concurrently. However, within the context of a single request, the processing is will be from RMS/ORFM to RTIL and then to the TaxWeb Tax Rules\*\*. In the event of a single request being very voluminous, this could result in slower throughput as it will be processed sequentially. Typical example of a large request could be a set of items ranged to large number of locations needing a taxation input.

To optimize the handling of large requests, threading is introduced in RTIL to enable concurrent processing. The approach of concurrent processing in RTIL is as follows:

- A large request is partitioned into a set of logical units of work called chunks.
- Each chunk is capable of being independently processed and the chunk is represented as a FisDocChnk object.
- Client (RMS/ORFM) partitions the large request into chunks and invokes RTIL.
- RTIL parses the requests and spawns the necessary threads to process the chunks individually.
- Concurrent requests are sent to the TaxWeb Tax Rules with the individual chunks and responses are received\*\*.
- Responses are tagged accordingly to support aggregation and sent back to the calling client (RMS/ORFM).

RMS/ORFM can trigger the concurrent processing in RTIL by setting a boolean flag called "Thread\_use" appropriately to Y/N which will be passed in the request. The behavior is controlled at the request level intentionally so that clients have the flexibility of optionally switching on concurrent processing in RTIL only for large request which merit concurrent processing for optimal throughput.

RTIL uses WebLogic WorkManager framework for thread management, which provides the following benefits:

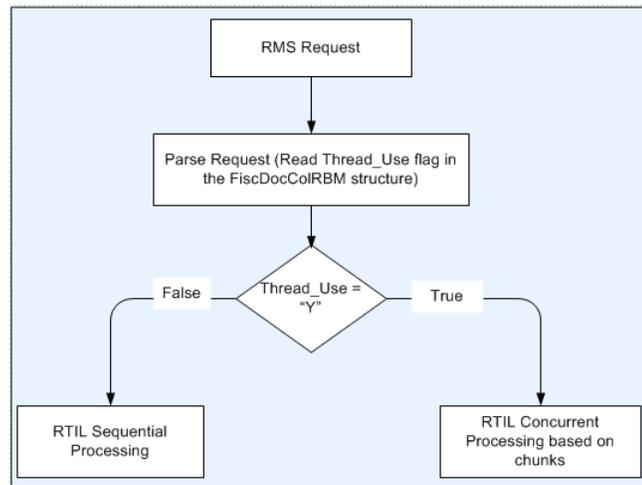
- Provides a concurrent programming API for EJBs/Servlets in a J2EE context.
- Provides a container managed alternative to java Thread API thereby enabling the container full visibility and control over the executing threads.
- Container based thread pooling and management.
- Standards based implementation

The configuration required for the work manager is detailed in the RTIL configuration and operational considerations section.

### RTIL Concurrent Flow Initiation

Figure 7-7 depicts how RTIL concurrent processing is triggered based on the Thread\_Use flag value contained in the RMS request.

**Figure 7-7 RTIL Concurrent Flow Initiation**

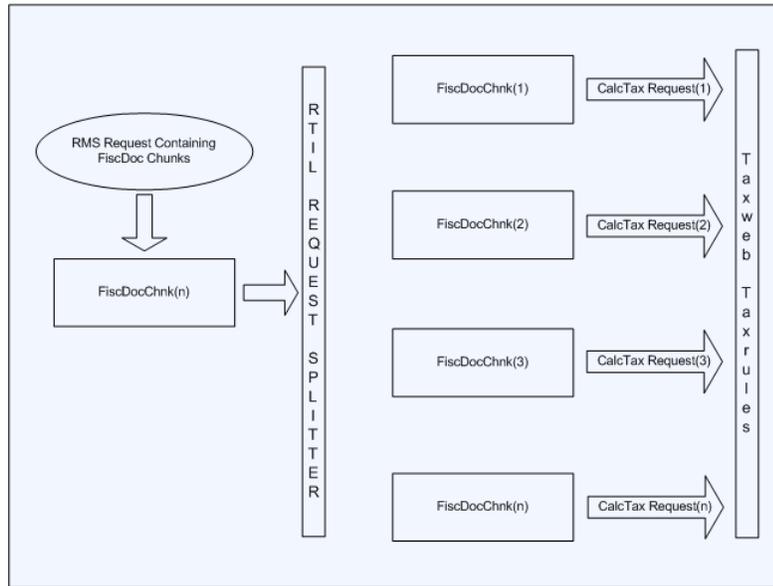


### RTIL Concurrent Request Processing Flow

Figure 7-8 depicts handling of a large RMS request in RTIL in a threaded fashion. The steps for handling large RMS request in a threaded fashion is as follows:

1. The RMS request contains the logical chunks (FiscDocChunk[n]).
2. RTIL splits it into individual chunks.
3. Each chunk is fired in parallel as an independent tax calculation request to the TaxWeb Tax Rules.
4. The responses are collected from the TaxWeb Tax Rules and tagged accordingly to support consolidation of the responses into a single aggregated response in RMS.

**Figure 7-8 RTIL Concurrent Request Processing Flow**



## Tax. Calculation Scenarios

For all the scenarios, the input parameters are based on the Retail Tax API objects. Only the tax objects are not used, and the detail data-mapping for the input parameters can change depending on each scenario.

### Scenario 1 - Inbound NF through User Entry

Enter the Inbound PO NF through the ORFM user interface. Once the NF is entered, tax call is initiated. The taxes returned by the external tax engine is compared against the informed taxes for any discrepancy.

### Scenario 2 - Inbound PO NFs through EDI

NFs from the supplier could be interfaced to RFM through EDI functionality. These NFs once generated also make a tax call through the batch provided. The taxes returned by the external tax engine is compared against the informed taxes for any discrepancy.

### Scenario 3 - PO Receiving

Tax calls are made from the RFM application during the receiving process based on the Resolution type on the Discrepancies identified during Validation of a NF.

### Scenario 4 - Outbound NF Issuing

For outbound transactions like Transfer, RTV and Inventory Adjustments initiated by the Retailer, ORFM generates the NFs automatically. These NFs generated makes the tax calls and the taxes are printed on the document before approval and inventory movement.

## **Scenario 4 - PO Tax Calculation**

The PO tax breakdown is the calculation of all applicable taxes of a PO. The PO information is similar to a fiscal document, and it is similar to the NF issuing process. The PO information utilizes the API format to have the tax calculated.

## **Scenario 5 - Item Creation**

During the item creation process, the sales taxes applicable to each location where the item is linked are calculated. The item creation process also makes a call to the tax API to obtain the sales tax rates. The same concept is applied to the default purchase taxes used for margin calculation.

Scenarios such as primary supplier change, and fiscal reclassification call the same tax integration defined for original item creation process.



---

---

## Working with RTIL

This chapter discusses the common issues that you have to consider as you progress towards a production environment involving the Oracle Retail Tax Integration Layer (RTIL). It is not a comprehensive list since they are very dependent on the retailer implementation.

### Operational Considerations

The operational issues that are to be considered when using RTIL are detailed in the following sub sections:

- [RTIL Log File Monitoring](#)
- [Log File Archive and Purge](#)
- [Time-out Configurations](#)
- [RTIL WorkManager Configurations](#)
- [General Troubleshooting for RTIL/TaxWeb Tax Rules](#)
- [Recovery in case of RTIL Compensation API failure](#)
- [RTIL and TaxWeb Tax Rules Outage Sequence/Start and Stop](#)
- [Probable Systemic Failure Scenarios in Tax Flows and Recovery](#)
- [Deploying RTIL](#)

### RTIL Log File Monitoring

Since RTIL is an integration subsystem that runs with no console, it is important to monitor the application log files that are created. The log files have to be monitored for the content (looking for exceptions), and also the size and growth of the exceptions. RTIL logs are created in the \$WEBLOGIC\_DOMAIN/log folder in the domain where RTIL is deployed.

The log files that are to be considered are:

- `rtil.log` – This is the primary log file which captures the RTIL server interactions with external tax engine. All the exceptions/failures that occur in the tax engine and RTIL interactions are logged in this file.
- `taxdatastage_services.log` – This log file captures RTIL interactions with RMS/ORFM application. All the exceptions/failures that happen during these interactions are logged in this file.

## Log File Archive and Purge

RTIL uses log4j for all of its logging control. It manages the logs size through its control file (log4j.properties), auto archive, and purge. This configuration file is embedded in the rtil-service.ear file in the lib\rtil-config.jar folder. This configuration file controls the behavior of logging in the rtil.log file. The file is configurable through the installer at install time. The file can also be manually modified for any changes that have to be made after the installation.

---

---

**Note:** Refer to the Apache Software Foundation <http://logging.apache.org/log4j/1.2/manual.html> for more details on log4j.

---

---

Below is a snapshot of the RTIL log4j.properties

```
log4j.appender.LOGFILE=org.apache.log4j.RollingFileAppender
log4j.appender.LOGFILE.MaxFileSize=5MB
log4j.appender.LOGFILE.File=./log/rtil.log
# Keep thirty backup files.
log4j.appender.LOGFILE.MaxBackupIndex=30
# Pattern to output: date priority [category] - message
log4j.appender.LOGFILE.layout=org.apache.log4j.PatternLayout
log4j.appender.LOGFILE.layout.ConversionPattern=%d %p [%c] - %m%n
```

The property **log4j.appender.LOGFILE.MaxFileSize** determines the maximum size of the log file before being rolled over. It is recommended to have a optimal size of not more than 5 MB. If the size of this file increases, it may lead to performance issues.

The property **log4j.appender.LOGFILE.MaxBackupIndex** depicts the maximum number of files to be archived in the \$WEBLOGIC\_DOMAIN/log folder. This value has to be based on the retailers' needs, the archival strategies, and the backup strategies.

## Time-out Configurations

Retail tax transactions involves interactions between multiple applications like the Oracle Retail Integration Bus (RIB), RMS, ORFM, any Store Inventory Management System (SIM), any Warehouse Management System (RWMS), RTIL and TaxWeb Tax Rules. Since there are multiple applications involved, they need to be handled very carefully in case of systemic failure in one or more applications.

The time-outs have to be configured diligently based on retailer's needs, volume of data, processing hardware etc. The time-outs have to ensure proper exits in case of slow responsive applications or systemic failures in applications.

A typical example of the retail tax transaction is as follows:

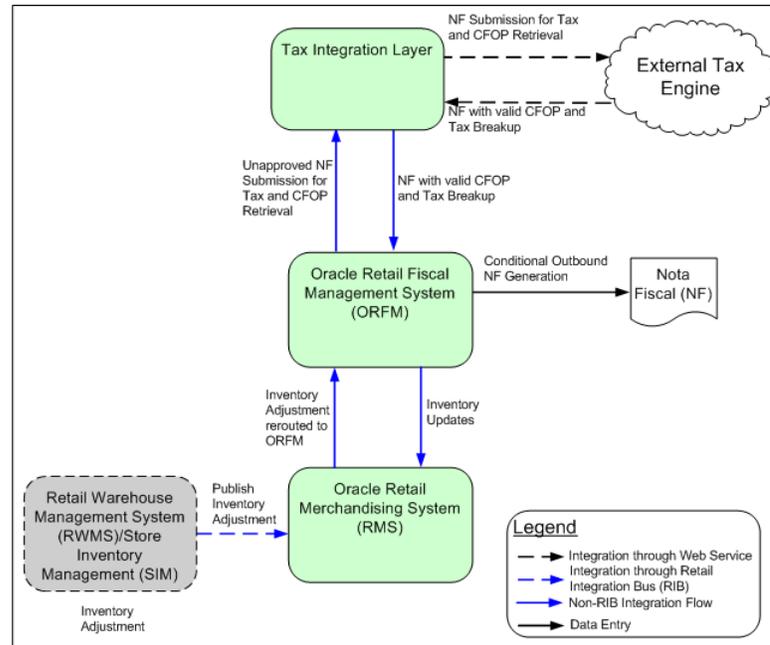
### Inventory Adjustment

An inventory adjustment retail transaction comprises of the following interactions between RIB, RMS, ORFM, RTIL and TaxWeb Tax Rules:

- Publication of Inventory Adjustment RIB message from any warehouse management system.
- RMS consumes the RIB message through the subscription API and routes it to ORFM.
- ORFM generates the Nota Fiscal (NF).

- During NF validation, ORFM invokes a synchronous tax call to a TaxWeb Tax Rules through RTIL.
- ORFM inventory is updated and the response is sent back to RIB.

**Figure 8–1 Inventory Adjustment Business Flow**



To ensure that the retail transactions are not adversely impacted in case of systemic failures in one or more applications, the time-outs has to be configured appropriately to have a exit path.

The general rule of thumb is that the time out configurations progressively decrease in a call sequence flow as we move from the top layer to the lowest layer, and have to be configured in the following subsystems:

- WebLogic Server
- RMS
- RIB

### URLInvoker Time-out

This time-out value is configured in the RMS table (RETAIL\_SERVICE\_REPORT\_TABLE) for the record (RS\_CODE=RTIL) and passed on to the RTIL client component (URLInvoker) when triggering a request to the RTIL server. Note the entry configured in the table is in the units of milliseconds. If the time-out value is not configured, then URLInvoker is set to a default value of 180 seconds.

This time-out caters to RTIL client component and ensures that the client component throws back an error response on expiry of the time-out in RTIL client or if the RTIL server fails to respond within the configured threshold.

### RIB Transaction Time-out

The RIB transactions are also configured with a time-out threshold value to ensure that RIB transactions do not wait forever.

While setting up the test environment, the following time-out values are configured:

- URLInvoker time-out – 1000 seconds
- RIB time-out – 1200 seconds

If any of the subsystems are responding slowly outside the range of its configured thresholds, time-outs are triggered. Eventually, these are sent back to RIB with an error status, which causes RIB to rollback the transaction and deliver it to the error hospital or the adapter will shut down based on the nature of the error.

---

---

**Note:** The time-out configurations specified are for indicative purpose only. They have to be customized based on the retailers' platform, environment, transactional volume and hardware.

---

---

## RTIL WorkManager Configurations

RTIL has a workmanager called "RTILWorkMgr" to support concurrent processing in RTIL for voluminous requests. The concurrent processing behavior is triggered in RTIL based on flag contained in its client's (RMS/ORFM) request.

Since RTIL uses WebLogic WorkManager, the following configurations depicted below can be modified/tuned in the WebLogic administrator console by a WebLogic Administrator:

- [Maximum Threads Constraint](#)
- [Capacity Constraint](#)

### Maximum Threads Constraint

This parameter depicts the maximum number of concurrent threads that can be allocated to execute requests. The default is set to 16. Typically, it is recommended to set this value based on the number of processors available in the system in which RTIL is deployed.

### Capacity Constraint

This parameter depicts the maximum limit of the requests that can be queued or executed. The default is set to 20000. After this, the WebLogic Server rejects requests.

---

---

**Note:** The work manager configurations specified are for indicative purpose only. They have to be customized based on the retailers' platform, environment, transactional volume, and hardware.

---

---

---

---

**Note:** In case the WebLogic Administrator wants to modify the workmanager configurations, the details are provided in [Working with RTIL](#).

---

---

## General Troubleshooting for RTIL/TaxWeb Tax Rules

This section discusses some of the errors (systemic or configuration related) that can occur in RTIL and its integration with the TaxWeb Tax Rules, and also RTIL integration with RMS/ORFM and consequently RIB. This section provides an overview of the errors, the cause and the corrective action that has to be taken. Some of the errors includes:

- RMS/ORFM is configured with an incorrect RTIL URL (Service Access Point)
- RTIL Service is not responding
- RTIL URL Configuration is correct but insufficient privileges in RMS database for RMS/ORFM to access RTIL
- Incomplete Deployment of TaxWeb Tax Rules libraries
- How do I know if RTIL is installed correctly?

### **RMS/ORFM is configured with an incorrect RTIL URL (Service Access Point)**

**Error Message** – In case the RTIL URL is incorrectly configured (format is wrong, hostname or port number is incorrect), any RMS/ORFM request to RTIL results in an error in the RTIL client component (URLInvoker hosted in RMS database schema), and the error message returned to the RMS is the incorrect URL format. The [Example 8-1](#) provides the incorrect URL format, with the incorrect part highlighted in **bold**.

#### **Example 8-1 Incorrect URL**

```
http://mspdv310.us.oracle.com:17069/rtil-web/invokeApp123?invocationType=CALC_
TAX&invocationKey=1234
```

**Cause** – The URL returned as depicted in the error message indicates that the RTIL URL is incorrectly configured in RMS.

**Resolution** – Update the URL column in the RETAIL\_SERVICE\_REPORT\_URL table with correct RTIL URL configuration as shown in [Example 8-2](#). The correction is made to the portion that is highlighted in bold in [Example 8-1](#).

#### **Example 8-2 Correct URL**

```
http://mspdv310.us.oracle.com:17069/rtil-web/invokeApp
```

### **RTIL Service is not responding**

**Error Message** – RTIL responds with the error message "Connection Refused" for any RMS/ORFM request to RTIL.

**Cause** – This error is typically thrown when the RTIL service is down.

**Resolution** – Restart the RTIL application instance from the WebLogic Administration Console. You must have WebLogic Admin privileges for this operation. In addition, an administrator has to investigate the WebLogic admin logs to ascertain the reason for the failure of the service before restarting it.

### **RTIL URL Configuration is correct but insufficient privileges in RMS database for RMS/ORFM to access RTIL**

**Error Message** – In ORA-00942: table or view does not exist ORA-06512: at "SYS.DBMS\_PICKLER", line 18ORA-06512: at "SYS.DBMS\_PICKLER", line 58ORA-06512: at line 1

**Resolution** – Usually occurs because the user used in the connection point does not have sufficient privileges to the Package or Objects being referred to in the application. Either change the user that is being used or make sure proper permissions and synonyms are created in Oracle.

## Incomplete Deployment of TaxWeb Tax Rules libraries

**Error Message** – Unknown Error occurred. Please check the log file for more details

The error in the RTIL logfile is provided in [Example 8-3](#)

### Example 8-3 RTIL Log File

```
java.lang.ClassNotFoundException: com.mcfox.tax.management.TaxRulesAPI
at
weblogic.utils.classloaders.GenericClassLoader.findLocalClass(GenericClassLoader.java:280)
at
weblogic.utils.classloaders.GenericClassLoader.findClass(GenericClassLoader.java:253)
at java.lang.ClassLoader.loadClass(ClassLoader.java:303)
at java.lang.ClassLoader.loadClass(ClassLoader.java:248)
at
weblogic.utils.classloaders.GenericClassLoader.loadClass(GenericClassLoader.java:177)
at java.lang.Class.forName0(Native Method)
at java.lang.Class.forName(Class.java:169)
at
com.oracle.retail.tax.servicegateway.MastersafPojoServiceGateway.init(Unknown Source)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:597)
```

**Cause** – This error typically occurs if the TaxWeb Tax Rules required libraries are not deployed in the \$WEBLOGIC\_DOMAN/lib folder of RTIL instance. The error message given is for reference only and will vary based on the missing library.

**Correction** – Refer to the TaxWeb Tax Rules Installation Guide and ensure that all the libraries (jar files) are made available in the \$WEBLOGIC\_DOMAIN/lib folder of the deployed RTIL instance. Before transferring the missing files, ensure that the RTIL application instance is shutdown from WebLogic admin console.

### How do I know if RTIL is installed correctly?

The following two steps are recommended for a quick sanity test to check if RTIL is deployed correctly and ready to serve requests:

- Technical Ping
- Application Ping

#### Technical Ping

This step is to ascertain if the RTIL service is running. Open a browser and enter the deployed RTIL URL in the address bar of the browser.

**Expected Result** – Enter the deployed RTIL URL in address bar of a browser `http://<hostname>:<portnum>/rtil-web/invokeApp` (For example `http://mspdv310.us.oracle.com:17069/rtil-web/invokeApp`). If the response in the browser window is "E| invocationKey not present", this implies RTIL service is running.

### Application Ping

This step is to ascertain if RTIL is correctly deployed and the configuration to RMS/ORFM is valid.

To sanity test an RTIL deployment, run the stored procedure provided in [Example 8-4](#) in an Oracle database schema in which RMS localized version has been deployed. Ensure that the arguments are replaced in the invokeURL call in the procedure to reflect the URL where RTIL is deployed.

#### Example 8-4 Procedure

```

Rem
Rem This SQL Script executes the Java Stored Procedure
Rem
SET SERVEROUTPUT ON
CALL dbms_java.set_output(2000);
DECLARE
  o_status VARCHAR2(255);
  o_error VARCHAR2(255);
BEGIN
--EX : invokeUrl
http://mspdv309.us.oracle.com:7003/rtil-web/invokeApp , 'CALC_
TAX', '1234', '180000',o_status, o_error);

  invokeUrl('<url>','<invocationType>', '<keyId>', '<timeout>',o_status, o_error);
  DBMS_OUTPUT.PUT_LINE('status ' || o_status);
  DBMS_OUTPUT.PUT_LINE('status ' || o_error);
END;
/

```

url = url where RTIL is deployed

(For example, 'http://mspdv309.us.oracle.com:7003/rtil-web/invokeApp')

invocationType = CALC\_TAX(for tax calls)

keyed = 1234 (a dummy key id)

time-out = 1800(a dummy time-out value)

**Expected Result** – 0L10N\_ROUTE\_INFO\_NOT\_FOUND@11234 – This would imply that RTIL is deployed correctly and also the configuration from RTIL to RMS/ORFM is valid.

## Recovery in case of RTIL Compensation API failure

Most of the retail tax transactions which have a taxation requirement are plain compute operations in TaxWeb Tax Rules without any data persistence. So failure in RTIL/RMS/ORFM post tax computations does not mandate a data rollback operation within TaxWeb Tax Rules. For certain exclusive ORFM scenarios like triangulation and recovery of ST, the TaxWeb Tax Rules performs persistence of data in its tables which mandate a rollback of data in case of failures in post processing.

For handling rolling back of persisted data in the TaxWeb Tax Rules tables in the event of a post processing failure in ORFM, it invokes a cancelTax (a compensation API) on to the TaxWeb Tax Rules which deals with explicit removal of persisted transaction data. In the rare event of failure of this compensation API, the failure will be logged in RTIL log file along with the data details. In such scenarios, a manual cleanup of records is required. The following tables have to be cleaned:

- TAX\_DOCFISCALHIST

- TAX\_ENQHIST
- TAX\_ENQITHIST
- TAX\_PESSOAHIST
- TAX\_REC\_ST\_ENTRADA
- TAX\_REC\_ST\_SAIDA

## RTIL and TaxWeb Tax Rules Outage Sequence/Start and Stop

The TaxWeb Tax Rules libraries have to be frequently updated since the content of TaxWeb Tax Rules changes due to incorporation of Brazil tax legislation changes.

RTIL, RMS, ORFM and RIB also would have patch upgrade requirements. To ensure smooth flow of operations at the customer environment, these subsystems have to be stopped and started in a particular order.

The recommended order for stopping the systems during a patch maintenance window is RIB, RMS/ORFM and RTIL and the start sequence is in the reverse order. RIB is an aggregation of distributed components and refer to *RIB Operations Guide*, for further details on start and stop of RIB internal components. TaxWeb Tax Rules being a library of RTIL does not have an independent operational start and stop sequence of its own.

## Probable Systemic Failure Scenarios in Tax Flows and Recovery

The following table provides the failure scenarios in tax flows:

Event	Scenario	How is it handled
RibForRMS goes down	RTIL, TaxWeb Tax Rules computation is completed, and RMS/ORFM tables are updated	Data will not be read/used as RIB transaction would have rolled back
RTIL failure	RIB transaction has updated RMS tables and RTIL stops responding	URLInvoker in RMS will receive an error as it is a synchronous call, which will result in an "E" status back to RIB. RIB then initiates a rollback of the data updated in RMS
RTIL fails to respond (very long running transaction)	RIB transaction has updated RMS tables and RTIL does not respond	URLInvoker will time-out resulting in an "E" status back to RIB which triggers a rollback of updated data in RMS
TaxWeb Tax Rules business failure	RIB transaction has updated RMS tables and RTIL fails to fetch the tax as the TaxWeb Tax Rules is unable to compute the tax due to a data issue in the request	RTIL returns an error to RMS and RMS sends back "E" status to RIB which triggers a rollback of updated data in RMS
RTIL failure in updating RMS tables with tax results	RIB transaction has updated RMS tables with request data, RTIL successfully calls the TaxWeb Tax Rules and obtains the taxes and RTIL updates to RMS tables results in a failure	RTIL returns an error to URLInvoker and this results in RMS sending an "E" to RIB which triggers a rollback of updated data in RMS

## TaxWeb Failures and Associated RIB Impacts

Some of the business flows will have a RIB XA transaction spawn across RIB, RMS, ORFM, RTIL, and TaxWeb. In such scenarios, the performance of a RIB transaction is inclusive and dependent on the participating downstream applications like RMS/ORFM/RTIL/TaxWeb. You need to be aware that any adverse behavior of TaxWeb due to poor performance in case of high volume, inherent bugs (if any), or any other issues can potentially result in RIB timeouts/in doubt transactions/RIB failures impacting the RIB adapters.

## Deploying RTIL

TaxWeb's lazy load solution is recommended if RTIL is deployed on a memory constrained environment.

The TaxWeb solution has the following two flavors primarily in terms of the memory footprint it occupies:

- Static Load Version
- Lazy Load Version

The Static Load Version loads all the tax rules available in the content at initialization time and has a larger memory footprint. This version requires about 5 GB of Java Heap Size space for it to be functional.

The Lazy Load Version implements an on-demand load feature. This implies only the tax rules needed to satisfy a tax computation request is loaded on the fly into the TaxWeb engine when it receives a tax calculation request.

The Lazy Load version is provided as an alternative to address deployments where in there is a crunch for physical memory.



## Understanding NF-e and SPED

This chapter provides details about NF-e and SPED and covers the following sections:

- [NF-e Solution](#)
- [Sistema Público de Escrituração Digital \(SPED\) Solution](#)

---

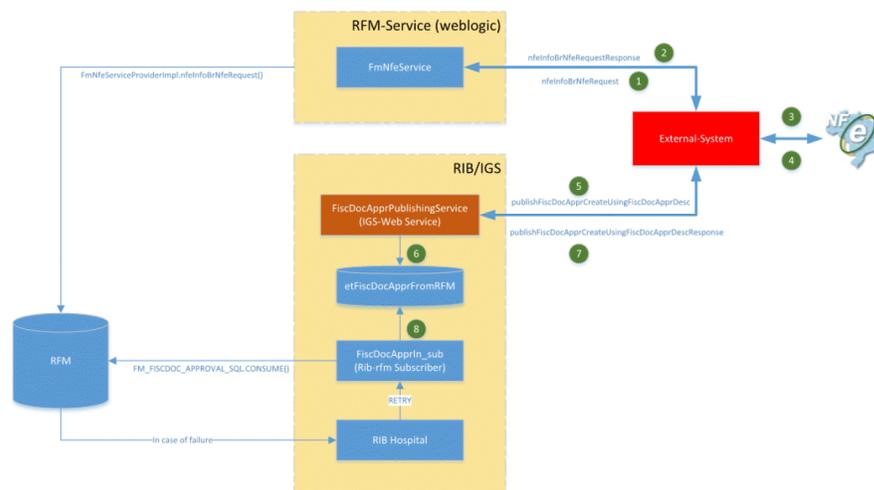
**Note:** ORFM 14.2 supports integration with fiscal partners for the issuing of NF-e (model 55), NFC-e (model 65). The NF-e integration uses a standard layout that can be leveraged by any specialist solution. For SPED, RFM provides a set of tables and views that have data from the entities managed by RFM (mainly NF related data). This integration is also standard; that is, not specific to any fiscal partner.

---

### NF-e Solution

The overall solution landscape considers that ORFM works with a third-party solution for NF-e generation and transmission to the government.

**Figure 9-1 NF-e Issuing Solution Approach**



This new approach has the following steps:

1. External-System request new NF by calling web-service operation `nfeIn-foBrNfeRequest` of `FmNfeService`.
2. RFM builds response with NF XML (not in Sefaz standard).

3. External-system builds NF-e XML and sends to SEFAZ.
4. External-system gets NF status with SEFAZ.
5. External-system calls web-service operation `publishFiscDocApprCreateUsing-FiscDocApprDesc` of `FiscDocApprPublishingService`.
6. RIB/IGS inserts new consume record into queue.
7. RIB/IGS builds response for service call.
8. RFM subscriber consumes new record and starts approval flow.

## NF-e Options

ORFM allows the generation of NF-e depending on the setup of the fiscal document type and fiscal document model type. The System options are available to default the document type to be used in transactions that have dependency in other modules such as RWMS or SIM.

Considering that the transactions where the NF-e issuing is applicable, like transfers, are initiated in RWMS. Hence the utilization code associated to each location must be the same utilization code set in `FM_SYSTEM_OPTIONS` as default utilization for outbound transactions, such as transfers, intercompany transfers, and RTVs. The default utilization code is used by RWMS to generate the NFS. This behavior is not controlled in the system and must be defined by the user. The utilization setup will allow users to define the document type and document model which drives the creation of an electronic NF or not.

## NF-e Events

The new NF-e events will need physical receiving confirmation. The new communication events for the NF-e receiving process are mandatory to all companies, since 2012. These events consist in:

- Acknowledgement of the operation
- Confirmation of the operation
- Information of operation not completed
- Ignorance of operation

The user needs to create the new NF-e integration points in order to address this requirement. NF-e integration is meant only for NF issuing processes and these new points will consider NF-e receiving processes (PO receiving).

Only the PO receiving NFs that have the access key field in the ORFM tables are considered for these new flows. That means only NF-e's received from vendors must have the physical receiving confirmation.

The NFs entered manually with the access key informed are also considered. The confirmation of the physical receiving or the cancelation of the receiving itself are sent to NF-e solution. The NF-e itself that comes from the vendor are already approved by SEFAZ during its issuing process.

Once the NF-e is inserted into ORFM (with access key), in case the physical receiving process does not even start and the user decides to not continue with the process for any reason, ORFM permits the deletion of the schedule and the associated NFs. This means the receiving of this vendor was canceled. A message will be sent to SEFAZ so the Government knows that the operation associated to the NF-e was not concluded.

In this way a message will be sent to NF-e integration with a reason for the rejection so it can be sent to the Government.

If the physical receiving has started, which means the operation is accepted at the retailer, the normal receiving process will continue until the physical receiving and NF approval.

After the NF approval, a confirmation will be sent to the NF-e integration. This confirmation would be required irrespective of the possible discrepancies in the receiving process.

## Webservices

In RFM NFe Integration, there are two synchronous Web-Services to communicate with the external systems and they attend the following specifications:

- All services are SOAP/HTTP based.
- All services comply with the JAX-WS specification.
- All services are WS-Addressing enabled.
- WS-Security can be plugged into these Web services without any code change.
- All Web services are Document Literal Wrapped.

They are deployed in different places since they have different functionalities.

**FmNfeService:** This service is deployed within the rfm-service.ear. It will be responsible for finding available NFs that should be sent to SEFAZ and provide a complete XML payload (with NF information) in the response for the calling system. The service has the following operations:

- **nfeInfoBrNfeRequest:** Provides the NF payload that should be sent to SEFAZ
- **ping:** It responds with the same data sent on the request. This service is used to check server availability.

### Valid values for tipo\_integração attribute

Attribute	Description
NFE_P	NF-e Approval request
NFE_C	NF-e Cancelation request
NFE_N	NF-e Nullification request
NFE_P_CO	NF-e Contingency Approval request
NFE_F	Contingency Approved NF-e waiting for Final SEFAZ Approval
NFE_S	NF-e Status Information request
NFE_A_P	NF-e event request - "Confirmation of Operation"
NFE_C_P	NF-e event request - "Operation not Performed"
NFE_U_P	NF-e event request - "Unknown Operation"

**FiscDocApprPublishingService:** This service is deployed with igs-service.ear (RIB). It will be responsible for enqueueing the payload sent from the calling system. The payload contains all necessary information for NF status consume process. This service has the following operations:

- **publishFiscDocApprCreateUsingFiscDocApprDesc:** This service will enqueue the payload in the request into etFiscDocApprFromRfm queue.
- **ping:** It responds with the same data sent on the request. This service is used to check server availability.

#### Attributes Description for FiscDocApprDesc

Attribute	Description
integration_id	Unique id of the integration process. This id was sent during FmNfeService.
fiscal_doc_id	Internal. Optional tag.
irl_doc_id	Internal. Optional tag.
sefaz_processing_date	SEFAZ processing date
status	Status of the SEFAZ processing
nfe_access_key	NFe Access key.
nfe_protocol	SEFAZ protocol.

#### Attributes Description for FiscDocApprDesc

Status	Description
A	NF-e Approved
DN	NF-e Denied (Denegado)
E	Request Processing Error: data or transmission errors (in any situation; emission, cancel or nullify)
A_C	NF-e Approved in Contingency mode
E_CO	Request new NF-e sequence
N_A	NF-e is successfully Nullified
C_A	NF-e is successfully Canceled
C_E	Cancel processing Error
A_F	The NF-e issued in Contingency is Approved
NFE_I	Intermediate flag that is updated by fiscal partner while the document has been sent to SEFAZ, just in case of emission.

## Sistema Público de Escrituração Digital (SPED) Solution

The SPED file that is generated by the fiscal partners contains a structure organized in blocks with opening and closure registers. The information in each block is as follows:

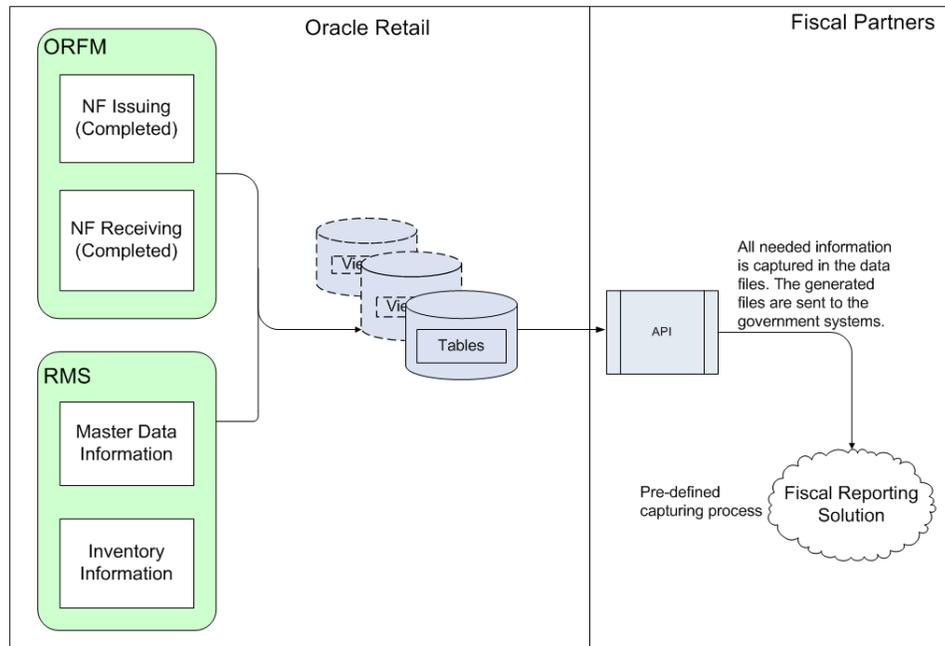
- Block 0: Opening, Identification and References
- Block C: Fiscal Documents I - Merchandise (ICMS/IPI)
- Block D: Fiscal Documents II - Services (ICMS)
- Block E: Fiscal counting of ICMS and IPI
- Block H: Physical Inventory
- Block K: Stock Position

- Block 1: Other information
- Block 9: Control and Closing of the Digital File

The file is generic and includes information pertinent to all types of companies. The retail segment is required to fill part of the entire file. In addition, the fiscal partner will be in charge of filling the mandatory information that is not provided by Oracle Retail.

The overall solution landscape is based on the existing views that integrate fiscal and master information with Oracle's fiscal partners.

**Figure 9–2 SPED Integration**



Because the SPED file has several sections corresponding to all types of transactions and fiscal data for a company, the scope of the integration (from the commercial system standpoint) was to make available all data kept within RMS and ORFM. Because of this, all types of data related to products for resale, and the fiscal transactions related to this type of product, are available in ORFM views and tables. Products used for consumption, assets, services and all transactions related to these types of product, are out of the scope for RMS/ORFM and are not available in the views.

The views and tables created to feed SPED include only data available in RMS/ORFM. The file is generated by the fiscal partner's solution, and fields (such as file opening and file closing), data related to the version of the SPED program, and all specific data for the file is provided by the fiscal partner. In addition, any field that can be deduced by the fiscal partner should also be provided by them.

SPED interfaces with a third-party system that shares the RMS database and opens the ports to establish network connectivity. It depends on the decision of the client to either host the SPED interfacing application (Interdados) within their environment or host it in a fiscal partner's environment. For security considerations, a separate schema should be created that contains only synonyms to as many objects required by the fiscal partner to generate the SPED information. Only the select privileges should be granted on these synonyms. No insert/update/delete should be allowed.

SPED interfaces are updated by batch import\_SPED.ksh. More details about this batch, see about it in this document.

## Block K - Stock Position

There is another type of information that will affect the Retailers Companies also related to stock of all the merchandise available to resale/commercialize. This info will take place in SPED accessory obligation (EFD-ICMS/IPI) and it is called Block K.

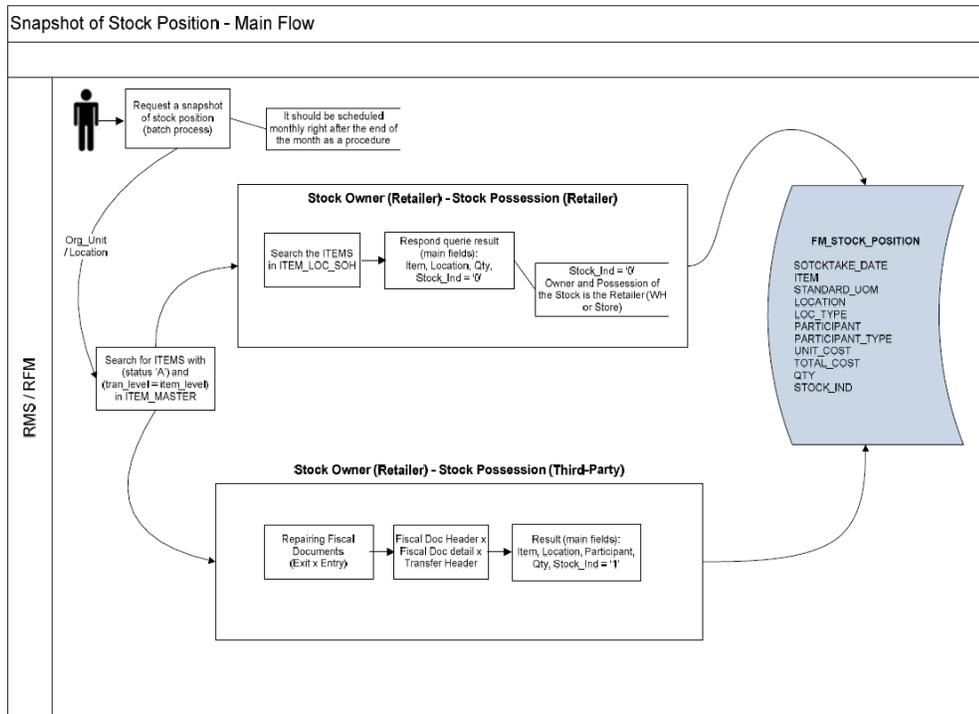
The Block K should be informed monthly and must cover the information about stock Position of all items type, a snapshot, right after the end of the month.

Until now, about the stock, only the Block H existed. This Block is different because in it should be informed the physical count quantity related to the stock of the Retailer and it is mandatory to inform once a year.

As a solution to fulfill this new legal requirement this document will detail a proposal of a new batch process that should cover most of the information required in the Block K. It is important to say that the main target of this Block K, are the Manufacture Industry. Hence, the Retailers will be affected in a specific point basically in one register called K200 (stock position).

In this register K200, is expected the Retailer inform not only the stock quantity owned by the Retailer, but also the stock owned by retailer that possession of this stock is in a third-party (External Finisher).

## Main Flow



## Performance Optimization for SPED

The anticipated volume of daily operations of a Tier 1 retailer are in the order of a few hundred thousand transactions per day. All these inbound and outbound transactions

create NFs that insert records into the ORFM prime tables. The SPED reporting feeds have a monthly frequency that lead to accumulation of large volumes of data. To mitigate the performance concerns for these large volumes, the solution approach has adopted the following strategy:

- Views only use all the foundation/static data tables.
- For the transaction tables that are bound to be voluminous, interface tables are created and populated with the necessary data, required for SPED by a batch job.
- The batch job is run on a daily basis so that the volumes are made manageable by operating on a day window instead of an entire month window.

The primary advantage of this solution is the high impact transaction tables are decoupled from direct access from the external system thereby avoiding the risk of slowing down the system during online operations. The batch job runs during the online shutdown window and the daily job operates on reduced volumes.



---

## Maintaining Tax Related Data - TaxWeb

There are a number of tables in RMS that contain data provided by (or derived from) an external tax engine (TaxWeb Tax Rules).

This chapter covers the following topics:

- [Installing Rules from TaxWeb Tax Rules](#)
- [Maintaining External Tax Information \(L10N\\_BR\\_T\\_EXTAX\\_MAINT\)](#)

### Installing Rules from TaxWeb Tax Rules

Perform the following steps to install an updated set of rules from a TaxWeb Tax Rules:

1. Stop the WebLogic Server for the domain in which RTIL is deployed.
2. Copy the TaxWeb Tax Rules .jar files to WEBLOGIC\_DOMAIN\_HOME/lib.
3. Start the WebLogic Server for the domain in which RTIL is deployed.
4. After a new TaxWeb .jar is loaded and the RTIL WebLogic Server runs, set the L10N\_BR\_EXTAX\_REFRESH\_CONFIG.REFRESH\_NEEDED to **Y**. This triggers updates to the tables in RMS that contain data provided by (or derived from) the external tax engine (like the TaxWeb Tax Rules)

### Item Cost for Non-Brazil Supplier

While computing the item cost, there is no need to make a tax call to tax rules engine if the supplier linked to an item is not a Brazilian Supplier. RMS checks the country of the supplier. RMS does not make a tax call to tax rules engine, and instead populates the item costing tables (ITEM\_COST\_HEAD and ITEM\_COST\_DETAIL) without any tax details for non-Brazilian suppliers.

There is no need to navigate to item costing screen in case supplier's country is not Brazil. The Negotiated Item Cost (NIC), Base Cost (BC), Extended Base Cost (EBC), and Inclusive Cost (IC) are calculated with zero tax values. The **Delivery country** is set to **Brazil**, and **primary delivery country** indicator is set to **Y**. The **NIC** static indicator is also set to **Y**. These defaults do not impact the cost calculations as there is no tax in this case.

RMS continues to make tax calls to the tax rules engine for Brazilian suppliers that is, suppliers for which the country on the supplier's primary address is set to Brazil. All values passed to tax rules engine for Cost computation must be in Brazilian Reais (BRL), irrespective of the supplier currency. If the supplier currency is BRL, then the values passed to tax rules engine and retrieved from tax rules engine are in BRL and can be stored in BRL in the RMS tables. In case the supplier operates in a currency other than BRL, then RMS converts the cost values into BRL using the currency

conversion rates defined in RMS. These converted values are passed to tax rules engine which return values of cost components and taxes in BRL. These cost and tax values are converted back into the supplier currency before storing them in the database.

Navigate to the item costing screen and click **Calculate Tax**. The data is retrieved with zero taxes.

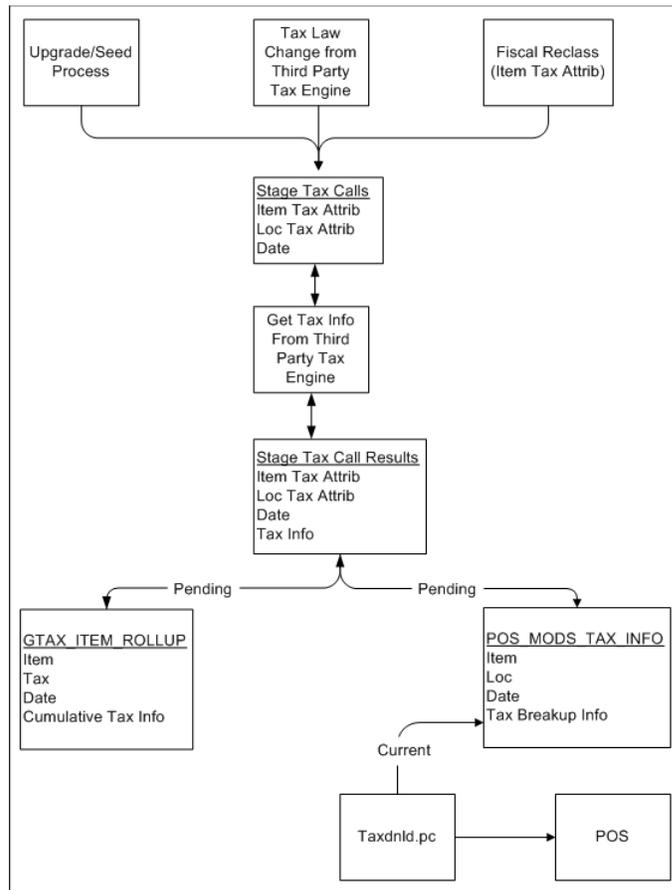
The data conversion scripts for non-Brazil supplier picks the items in Approved status and inserts the data as it is from dat files to tables.

The item costing will not be made, and it is not a correct business scenario, when an item has a non-Brazil delivery country.

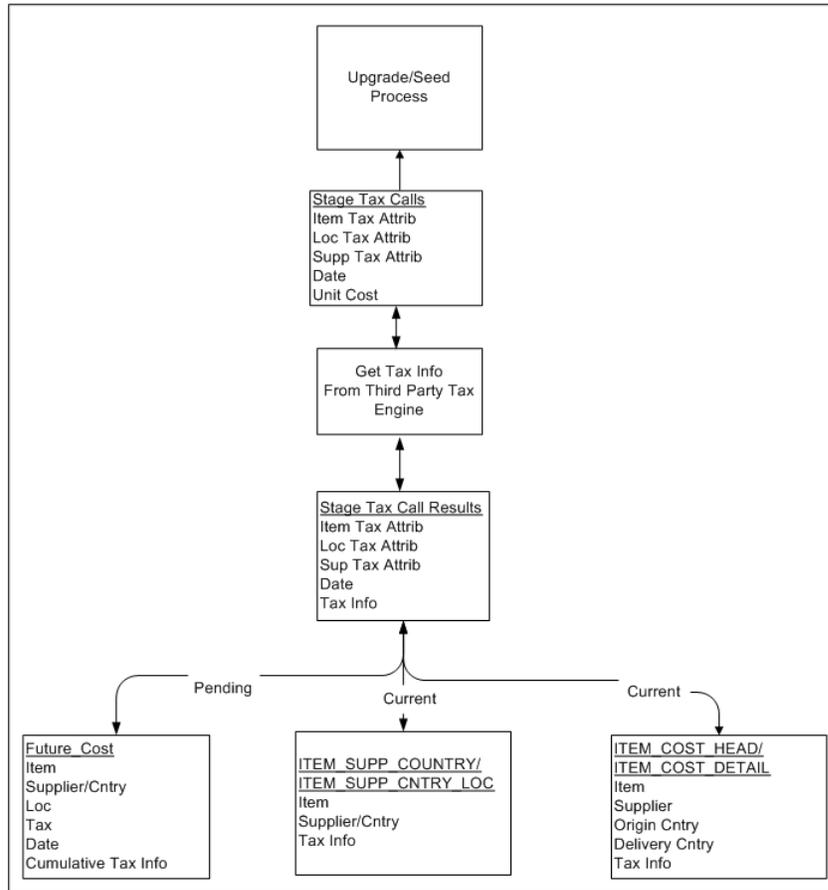
## Program Flow

The basic flow of data from a triggering event to the RMS tables holding tax information is shown below. There are three diagrams, one for retail and two for cost.

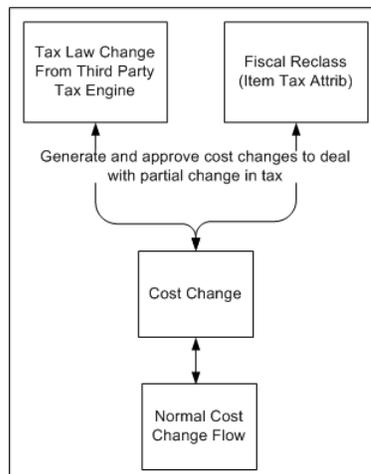
**Figure 10–1 Retail Tax Flow**



**Figure 10-2 Cost Tax Flow**



**Figure 10-3 Cost Tax Flow Maintenance**



## Maintaining External Tax Information (L10N\_BR\_T\_EXTAX\_MAINT)

The tables can be divided into two categories depending on whether they hold data related to retail tax or cost tax.

- Retail Tables – GTAX\_ITEM\_ROLLUP and POS\_MODS\_TAX\_INFO
- Cost Tables – ITEM\_SUPP\_COUNTRY, ITEM\_SUPP\_COUNTRY\_LOC, FUTURE\_COST, ITEM\_COST\_HEAD, ITEM\_COST\_DETAIL

These tables can also be categorized by whether they hold the current state tax information or pending state tax information. The current state tables only get updated when the transaction effecting tax becomes active and that pending state tables get updated as soon as the transaction effecting tax becomes known. For example, when we understand from the TaxWeb Tax Rules that the sales tax is moving from 10% to 12% in San Paulo in two weeks, the pending state tables are updated as soon this information is known. The current state tables are not updated until two weeks from now when the rate change actually occurs.

- Current State Tables – ITEM\_SUPP\_COUNTRY, ITEM\_SUPP\_COUNTRY\_LOC, ITEM\_COST\_HEAD, ITEM\_COST\_DETAIL
- Pending State Tables – GTAX\_ITEM\_ROLLUP, POS\_MODS\_TAX\_INFO, FUTURE\_COST

Since the tax information is stored in RMS tables, these tables have to contain the factors that determine how tax change is calculated.

The following factors drive retail tax recalculations:

- Item tax attributes (including dimensions)
- Location tax attributes/location address
- Date

The following factors drive cost tax recalculations:

- Item tax attributes (including dimensions)
- Location tax attributes/location address
- Supplier tax attributes/supplier address
- Unit cost
- Date

L10N\_BR\_T\_EXTAX\_MAINT\_SQL uses data grouping to minimize the number of calls to an external tax engine.

The idea behind grouping is that it is the fiscal attributes that determine what the tax is, not the particular entity (many different items may all share the same fiscal attributes - same thing for suppliers and locations). The ITEM\_LOC table might have 1 million rows on it, but there might only be 125,000 unique combinations of item fiscal attributes and location fiscal attributes between them. If that is the case, only 125,000 calls need to be made to the external tax engine instead of 1 million calls.

---

---

## Maintaining Tax Related Data - TaxWeb

The batch overview provided in this chapter identifies the functional area description with the batch processes illustrated in the designs. This overview allows you to determine how a business function works behind the scenes.

Batch designs describe how, on a technical level, an individual batch module works and the database tables that it affects. In addition, batch designs contain file layout information that is associated with the batch process.

This chapter covers the following topics:

- [TaxWeb Tax Rules Integration and Maintaining Batches](#)
  - fmtaxchg.pc
  - fmtaxupld.pc
  - l10n\_br\_exec\_tax\_recalc.ksh
  - l10n\_br\_fiscal\_item\_reclass\_cost.ksh
  - l10n\_br\_fiscal\_loc\_reclass\_cost.ksh
  - l10n\_br\_fiscal\_loc\_reclass\_retail.ksh
  - l10n\_br\_fiscal\_reclass\_item\_finish\_retail.ksh
  - l10n\_br\_fiscal\_reclass\_item\_process\_retail.ksh
  - l10n\_br\_fiscal\_reclass\_item\_setup\_retail.ksh
  - l10n\_br\_refresh\_extax\_finish\_retail.ksh
  - l10n\_br\_refresh\_extax\_future\_cost.ksh
  - l10n\_br\_refresh\_extax\_process\_retail.ksh
  - l10n\_br\_refresh\_extax\_setup\_retail.ksh
  - l10nbr\_taxweb\_fisdnlld.pc
- [Financial Posting Batches](#)
  - fmtrandata.pc
  - fmfinpost.pc
  - import\_SPED.pc
- [Deal Income Postings Batch](#)
  - fmdealinc.pc
- [ORFM Utility Batches](#)

- fm\_async\_queue\_cleanup.ksh
- fm\_batch\_avg\_sales\_price.ksh
- fm\_batch\_freclass\_sql.ksh
- fm\_generate\_nf\_stkcount.ksh
- fm\_reprocess\_nf\_status\_f.ksh
- fm\_stock\_position.ksh
- fmedinf.pc
- fmmanifest.ksh
- fmprepost.pc
- l10n\_ld\_iindfiles.ksh
- [Localization Purge Batch Process](#)
  - fm\_batch\_clear\_calc\_stg\_t.ksh
  - fmpurge.pc
  - l10nbrfreclsprg.pc
  - l10npurge.pc

## TaxWeb Tax Rules Integration and Maintaining Batches

### fmtaxchg (Tax Changing Process)

<b>Module Name</b>	fmtaxchg.pc
<b>Description</b>	
<b>Functional Area</b>	Tax changing Process
<b>Module Type</b>	
<b>Module Technology</b>	ProC
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

#### Design Overview

Use this batch program to process the item/location records affected by tax change.

The fmtaxchg RFM batch processes the records from FM\_TAXCHG\_HEADER and FM\_TAXCHG\_DETAIL tables with 'U' un-processed or 'E' error status.

The major functionality of this batch program is divided into the following three steps:

1. Making a call to RMS packaged function L10N\_BR\_WACADJ\_SQL.WAC\_UPDATE to update WAC and TRAN\_DATA postings for cost variance. In case of unhandled exceptions and package errors, the batch stops further processing and an error is reported in error file.

2. Making a call to FM\_T\_EXT\_TAXES\_SQL and FM\_S\_EXT\_TAXES\_SQL with taxchg\_ids for locations having control\_rec\_st\_ind set to 'Y' in V\_FISCAL\_ATTRIBUTES, to update the ST history in MTR tables. In case of unhandled exceptions and package errors, the batch stops further processing and an error is reported in error file.

The STATUS column is updated to 'A' in FM\_TAXCHG\_HEADER table of WAC and ST history tables if the records are updated successfully.

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Ad hoc
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	Multi-Threaded based on taxchg_id

### Restart/Recovery

The logical unit of work is taxchg\_id.

### Key Tables Affected

Tables	Select	Insert	Update	Delete
FM_TAXCHG_HEADER	Yes	No	Yes	No
FM_TAXCHG_DETAIL	Yes	No	No	No
V_FM_RESTART_TAXCHG	Yes	No	No	No
V_FISCAL_ATTRIBUTES	Yes	No	No	No

## fntaxupld (Tax Change Upload)

<b>Module Name</b>	fntaxupld.pc
<b>Description</b>	
<b>Functional Area</b>	WAC and Substituição Tributária (ST) history update
<b>Module Type</b>	
<b>Module Technology</b>	ProC
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

## Design Overview

Use this batch program to upload the 'Tax Change Upload' input file consisting of tax rule changes into the system for processing purpose.

FMTAXUPLD.PC is a Pro\*C program that runs as a module in Adhoc. The purpose is to upload and process Tax Change input file consisting of tax rule changes. The module accepts Tax Change data contained in a flat file (ASCII text) and formatted to match the prescribed retail input file format.

All items are validated, FMTAXUPLD.PC writes a row into the FM\_STG\_TAXCHG\_DTL table.

## Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Ad hoc
Pre-Processing	N/A
Post-Processing	Fmtaxupld post
Threading Scheme	N/A

## Restart/Recovery

Since this is a file-based upload, file-based restart/recovery logic is applied. The commit\_max\_ctr field should be set to prevent excessive rollback space usage, and to reduce the overhead of file I/O.

## Key Tables Affected

Tables	Select	Insert	Update	Delete
FM_STG_TAXCHG_DTL	No	Yes	No	Yes
ITEM_MASTER	Yes	No	No	No
ITEM_LOC_SOH	Yes	No	No	No
SYSTEM_OPTIONS	Yes	No	No	No
WH	Yes	No	No	No

## Input File

The input file name is not fixed, and it is determined by a run time parameter. Records rejected by the import process are written to a reject file. The reject file name is not fixed, and it is determined by a run time parameter.

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	File record descriptor	Char (5)	FHEAD	It describes the file line type.
	Line number	Number (10)	0000000001	it is a sequential file line number.

Record Name	Field Name	Field Type	Default Value	Description
	File Type	Char (4)	TXUP	It is the Tax change Upload.
	File create date	Char (14)		It is YYYYMMDDHH24MISS format.
FDETL	File record descriptor	Char(5)	FDETL	It describes the file line type.
	Line number	Number (10)		It is the sequential file line number.
	Action Type	Char (1)		Following is the type of mode: 'Insert' or 'Delete'
	NF number	Number (15)		The NF number is used only for delete scenario.
	Series No	Char (20)		The series number is used only for delete scenario.
	Item	Char (25)		It is the item number.
	Location_id	Number (10)		It is the location ID.
	Location_type	Char (1)		Following is the type of location: 'Store' or 'Warehouse'
	Quantity	Number (12)		Quantity * 10000 (4 implied decimal places), quantity of item at that location. It is used to update the MTR history tables.
	Unit cost	Number (20)		Unit cost * 10000 (4 implied decimal places), Unit cost of the item. It is assumed that it will be in location currency.
	ICMS unit amount	Number (20)		Imposto Sobre Circulação de Mercadorias e Serviços (ICMS) unit amount * 10000 (4 implied decimal places), ICMS amount changed per unit. It is assumed that it will be in location currency.
	ICMSST unit amount	Number (20)		ICMSST unit amount * 10000 (4 implied decimal places), ICMSST amount changed per unit. It is assumed that it will be in location currency.
	ICMSST Base Value	Number (20)		ICMSST Base Value* 10000 (4 implied decimal places), ICMSST Base value per unit for the item. It is assumed that it will be in location currency.

Record Name	Field Name	Field Type	Default Value	Description
	ICMSSTE unit amount	Number (20)		ICMSSTE unit amount * 10000 (4 implied decimal places), ICMSSTE amount changed per unit. It is assumed that it will be in location currency
	ICMSSTE Base Value	Number (20)		ICMSSTE Base Value * 10000 (4 implied decimal places), ICMSSTE Base value per unit for the item. It is assumed that it will be in location currency.
	New average cost	Number (20)		New average cost * 10000 (4 implied decimal places), New average cost of the item after tax law change. It is assumed that it will be in location currency.
FTAIL	File record descriptor	Char (5)	FTAIL	It describes the file record type.
	Line number	Number (10)		It is a sequential file line number (total number of lines in the file).
	Number of detail records	Number (10)		It is the number of FDETL lines in the file.

## l10n\_br\_exec\_tax\_recalc

<b>Module Name</b>	l10n_br_exec_tax_recalc.ksh
<b>Description</b>	
<b>Functional Area</b>	TaxWeb Integration
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This batch job deals with executing fiscal entity reclass transactions for that day. This logic is performed on the effective date of the fiscal entity reclassification. This consists of updating the cost fields on ITEM\_SUPP\_COUNTRY\_LOC, ITEM\_SUPP\_COUNTRY and updating the localization extension tables with the new fiscal attribute values.

The work is done in the L10N\_BR\_T\_EXEC\_RECLASS\_SQL package.

## Scheduling Constraints

Schedule Information	Description
Processing Cycle	Daily
Scheduling Considerations	Phase 3
Pre-Processing	l10n_br_fiscal_item_reclass_cost.ksh
Post-Processing	sccext.pc
Threading Scheme	Not applicable

## Restart/Recovery

Not applicable

## Key Tables Affected

Tables	Select	Insert	Update	Delete
COST_TAX_RECALC	Yes	No	Yes	Yes
ITEM_COST_HEAD	Yes	No	No	No
COUNTRY_ATTRIB	Yes	No	No	No
MV_L10N_ENTITY	Yes	No	No	No
ITEM_SUPP_COUNTRY_LOC	Yes	No	Yes	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
STORE_L10N_EXT	Yes	Yes	Yes	No
L10N_BR_T_RECLASS_ENTITY	Yes	No	No	No
WH_L10N_EXT	Yes	Yes	Yes	No
PARTNER_L10N_EXT	Yes	Yes	Yes	No
SUPS_L10N_EXT	Yes	Yes	Yes	No
L10N_BR_RECLASS_ENTITY_NV	Yes	No	No	No
L10N_ATTRIB	Yes	No	No	No
L10N_ATTRIB_GROUP	Yes	No	No	No
EXT_ENTITY	Yes	No	No	No
ITEM_COST_HEAD	Yes	Yes	Yes	No
ITEM_COST_DETAIL	Yes	Yes	Yes	Yes

## l10n\_br\_fiscal\_item\_reclass\_cost

Module Name	l10n_br_fiscal_item_reclass_cost.ksh
Description	

<b>Functional Area</b>	TaxWeb Integration
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This module creates cost changes for all item/supplier/origin country/location combinations that are affected by fiscal item reclassifications. The normal cost change processes will then take care of updating RMS with the new tax law information when necessary. If the default location is not ranged to an item being reclassified, the ITEM\_COST\_HEAD and ITEM\_COST\_DETAIL tables are handled directly.

This batch stores and maintains data that comes from the external tax engine like the TaxWeb Tax Rules.

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	

### Restart/Recovery

Not applicable

### Key Tables Affected

Table	Select	Insert	Update	Delete
GTAX_COST_CHANGE_GTT	Yes	Yes	Yes	No
L10N_BR_FISCAL_RECLASS	Yes	No	No	No
ITEM_SUPP_COUNTRY_LOC	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
ADDR	Yes	No	No	No
FUTURE COST TABLES	Yes	Yes	Yes	Yes
ITEM_COST_HEAD	Yes	Yes	Yes	No
MV_L10N_ENTITY	Yes	No	No	No
COUNTRY_ATTRIB	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No

Table	Select	Insert	Update	Delete
ITEM_COST_DETAIL	Yes	Yes	Yes	No
SYSTEM_OPTIONS	Yes	No	No	No
SUPS	Yes	No	No	No
COST_SUSP_SUP_HEAD	No	Yes	No	No
COST_SUSP_SUP_DETAIL_LOC	No	Yes	No	No
COST_EVENT_RUN_TYPE_CONFIG	Yes	No	No	No
COST_EVENT	No	Yes	No	No
ITEM_COST_DETAILS	No	No	No	Yes
STORE	Yes	No	No	No
WH	Yes	No	No	No
partner	Yes	No	No	No
L10N_TAX_OBJECT_CONFIG	Yes	No	No	No
VAT_ITEM	Yes	No	No	No

## l10n\_br\_fiscal\_loc\_reclass\_cost

<b>Module Name</b>	l10n_br_fiscal_loc_reclass_cost.ksh
<b>Description</b>	
<b>Functional Area</b>	TaxWeb Integration
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This module deals with processing cost logic of an entity fiscal entity reclassification. This consists of identifying the effected item/supp/country/loc rows, creating the new cost\_tax\_recalc records and sending them to the cost engine. This logic is preformed when the fiscal entity reclass is initially created.

The work is done in the L10N\_BR\_T\_EXTAX\_MAINT\_SQL.FISCAL\_LOC\_RECLASS\_COST package.

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Daily
Scheduling Considerations	Phase 1

Schedule Information	Description
Pre-Processing	l10n_br_fiscal_loc_reclass_retail.ksh
Post-Processing	None
Threading Scheme	N/A

### Restart/Recovery

Not applicable

### Key Tables Affected

Tables	Select	Insert	Update	Delete
COST_TAX_RECALC	Yes	Yes	No	No
ITEM_COST_HEAD	Yes	No	No	No
COUNTRY_ATTRIB	Yes	No	No	No
MV_L10N_ENTITY	Yes	No	No	No
ITEM_SUPP_COUNTRY_LOC	Yes	No	Yes	No
ITEM_SUPP_COUNTRY	No	No	No	No
L10N_BR_T_RECLASS_ENTITY	Yes	No	Yes	No
COST_EVENT	No	Yes	No	No

## l10n\_br\_fiscal\_loc\_reclass\_retail

<b>Module Name</b>	l10n_br_fiscal_loc_reclass_retail.ksh
<b>Description</b>	
<b>Functional Area</b>	TaxWeb Integration
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This module allows for recalculation of tax based on the changed fiscal attribute for store/warehouse/partner/supplier entities.

This consists of identifying the effected item/loc rows, calling the tax engine, and updating POS\_MODS\_TAX\_INFO and GTAX\_ITEM\_ROLLUP. This logic is performed when the fiscal entity reclass is initially created.

The work is done in the L10N\_BR\_T\_EXTAX\_MAINT\_SQL.FISCAL\_LOC\_RECLASS\_EXTAX\_SETUP package.

## Scheduling Constraints

Schedule Information	Description
Processing Cycle	Daily
Scheduling Considerations	Phase 1
Pre-Processing	None
Post-Processing	l10n_br_fiscal_loc_reclass_cost.ksh
Threading Scheme	N/A

## Restart/Recovery

Not applicable

## Key Tables Affected

Tables	Select	Insert	Update	Delete
L10N_BR_FISCAL_RECLASS	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
L10N_BR_T_RECLASS_ENTITY	Yes	No	No	No
L10N_BR_T_EXTAX_STG_RETAIL	No	Yes	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
GTAX_ITEM_ROLLUP	Yes	Yes	No	No
ITEM_SUPP_COUNTRY_DIM	Yes	No	No	No

## l10n\_br\_fiscal\_reclass\_item\_finish\_retail

<b>Module Name</b>	l10n_br_fiscal_reclass_item_finish_retail.ksh
<b>Description</b>	
<b>Functional Area</b>	TaxWeb Integration
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

## Design Overview

This module picks up the group level tax call results that l10n\_br\_fiscal\_reclass\_item\_process\_retail.ksh places on L10N\_BR\_T\_EXTAX\_RES\_RETAIL and L10N\_BR\_T\_

EXTAX\_RES\_RETAIL\_DET and explodes them back to the item/location level. It then uses the item/location level information to write tax information to GTAX\_ITEM\_ROLLUP and POS\_MODS\_TAX\_INFO tables.

This batch stores and maintains data that comes from the external tax engine like the TaxWeb Tax Rules.

The work is done in the L10N\_BR\_T\_EXTAX\_MAINT\_SQL package.

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Phase 1
Pre-Processing	l10n_br_fiscal_reclass_item_process_retail.ksh
Post-Processing	l10n_br_fiscal_item_reclass_cost.ksh
Threading Scheme	Not applicable

### Restart/Recovery

Not applicable

### Key Tables Affected

Tables	Select	Insert	Update	Delete
L10N_BR_T_EXTAX_HELP_RTL_GTT	Yes	Yes	No	No
ITEM_LOC	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ITEM_SUPP_COUNTRY_DIM	Yes	No	No	No
L10N_BR_T_EXTAX_RES_RETAIL	Yes	No	No	No
L10N_BR_FISCAL_RECLASS	Yes	No	Yes	No
L10N_BR_T_EXTAX_DEST_GRP_HELP	Yes	No	No	No
V_BR_ITEM_FISCAL_ATTRIB	Yes	No	No	No
L10N_BR_EXTAX_HELP_NV_PAIR	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
GTAX_ITEM_ROLLUP	No	Yes	Yes	No
POS_MODS_TAX_INFO	No	Yes	Yes	No

Tables	Select	Insert	Update	Delete
L10N_BR_T_EXTAX_RES_RETAIL_DET	Yes	No	No	No
L10N_BR_FISCAL_RECLASS_NV	Yes	No	No	No
L10N_ATTRIB	Yes	No	No	No
L10N_ATTRIB_GROUP	Yes	No	No	No
EXT_ENTITY	Yes	No	No	No
ITEM_COUNTRY_L10N_EXT	No	Yes	Yes	No

## l10n\_br\_fiscal\_reclass\_item\_process\_retail

<b>Module Name</b>	l10n_br_fiscal_reclass_item_process_retail.ksh
<b>Description</b>	
<b>Functional Area</b>	TaxWeb Integration
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This module picks the staged groups placed on L10N\_BR\_T\_EXTAX\_STG\_RETAIL by l10n\_br\_fiscal\_reclass\_item\_setup\_retail.ksh and calls the external tax provider with them. The results of these calls are placed in the L10N\_BR\_T\_EXTAX\_RES\_RETAIL and L10N\_BR\_T\_EXTAX\_RES\_RETAIL\_DET tables.

This batch stores and maintains data that comes from the external tax engine like the TaxWeb Tax Rules.

The work is done in the L10N\_BR\_T\_EXTAX\_MAINT\_SQL package.

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Phase 1
Pre-Processing	l10n_br_fiscal_reclass_item_setup_retail.ksh
Post-Processing	l10n_br_fiscal_reclass_item_finish_retail.ksh
Threading Scheme	Not applicable

### Restart/Recovery

Not applicable

## Key Tables Affected

Tables	Select	Insert	Update	Delete
RETAIL_SERVICE_REPORT_URL	Yes	No	No	No

## l10n\_br\_fiscal\_reclass\_item\_setup\_retail

<b>Module Name</b>	l10n_br_fiscal_reclass_item_setup_retail.ksh
<b>Description</b>	
<b>Functional Area</b>	TaxWeb Integration
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This module looks at all the valid item/location combinations in RMS for items having their fiscal attributes changed. It determines the unique fiscal attribute groups that cover all the item/locations combinations and puts the groups on a stage table (L10N\_BR\_T\_EXTAX\_STG\_RETAIL).

This batch stores and maintains data that comes from the external tax engine like the TaxWeb Tax Rules.

The work is done in the L10N\_BR\_T\_EXTAX\_MAINT\_SQL package.

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Phase 1
Pre-Processing	None
Post-Processing	l10n_br_fiscal_reclass_item_process_retail.ksh
Threading Scheme	Not applicable

### Restart/Recovery

Not applicable

## Key Tables Affected

Tables	Select	Insert	Update	Delete
L10N_BR_T_EXTAX_STG_ RETAIL	Yes	Yes	Yes	Yes
L10N_BR_T_EXTAX_RES_ RETAIL	No	No	No	Yes
L10N_BR_T_EXTAX_RES_ RETAIL_DET	No	No	No	Yes
V_BR_T_STORE	Yes	No	No	No
L10N_BR_ENTITY_CNAE_ CODES	Yes	No	No	No
V_BR_T_PARTNER	Yes	No	No	No
V_BR_T_WH	Yes	No	No	No
WH	Yes	No	No	No
L10N_BR_SUP_TAX_REGIME	Yes	No	No	No
ADDR	Yes	No	No	No
COUNTRY_TAX_JURISDICTION	Yes	No	No	No
COUNTRY	Yes	No	No	No
STATE	Yes	No	No	No
L10N_BR_EXTAX_HELP_NV_ PAIR	Yes	Yes	Yes	Yes
L10N_BR_T_EXTAX_DEST_GRP_ HELP	Yes	Yes	Yes	Yes
EXT_ENTITY	Yes	No	No	No
L10N_ATTRIB_GROUP	Yes	No	No	No
L10N_ATTRIB	Yes	No	No	No
L10N_BR_FISCAL_RECLASS	Yes	Yes	No	No
ITEM_LOC	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ITEM_SUPP_COUNTRY_DIM	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
L10N_BR_T_TAX_CALL_STAGE_ ROUTE	Yes	No	No	No
V_BR_ITEM_FISCAL_ATTRIB	Yes	No	No	No
V_BR_T_SUPS	Yes	No	No	No
V_BR_ITEM_COUNTRY_ ATTRIBUTES	Yes	No	No	No

## l10n\_br\_refresh\_extax\_finish\_retail

<b>Module Name</b>	l10n_br_refresh_extax_finish_retail.ksh
<b>Description</b>	
<b>Functional Area</b>	TaxWeb Integration
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This module picks the group level tax call results that l10n\_br\_refresh\_extax\_process\_retail.ksh places on L10N\_BR\_T\_EXTAX\_RES\_RETAIL and L10N\_BR\_T\_EXTAX\_RES\_RETAIL\_DET tables and explodes them back to the item/location level. It then uses the item/location level information to write tax information to GTAX\_ITEM\_ROLLUP and POS\_MODS\_TAX\_INFO tables.

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Phase 1
Pre-Processing	l10n_br_refresh_extax_process_retail.ksh
Post-Processing	None
Threading Scheme	Not applicable

### Restart/Recovery

Not applicable

### Key Tables Affected

Tables	Select	Insert	Update	Delete
L10N_BR_EXTAX_REFRESH_CONFIG	No	No	Yes	No
L10N_BR_T_EXTAX_HELP_RTL_GTT	Yes	Yes	No	No
ITEM_LOC	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ITEM_SUPP_COUNTRY_DIM	Yes	No	No	No

Tables	Select	Insert	Update	Delete
L10N_BR_T_EXTAX_RES_ RETAIL	Yes	No	No	No
L10N_BR_T_EXTAX_DEST_GRP_ HELP	Yes	No	No	No
V_BR_ITEM_FISCAL_ATTRIB	Yes	No	No	No
L10N_BR_EXTAX_HELP_NV_ PAIR	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
GTAX_ITEM_ROLLUP	No	Yes	Yes	No
POS_MODS_TAX_INFO	No	Yes	Yes	No
L10N_BR_T_EXTAX_RES_ RETAIL_DET	Yes	No	No	No
V_BR_ITEM_COUNTRY_ ATTRIBUTES	Yes	No	No	No

## l10n\_br\_refresh\_extax\_future\_cost

<b>Module Name</b>	l10n_br_refresh_extax_future_ cost.ksh
<b>Description</b>	
<b>Functional Area</b>	TaxWeb Integration
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This module obtains a list of fiscal attributes that have tax law changes scheduled against them since its last run. It then uses this information to determine what item/supplier/origin country/locations are affected by those scheduled tax law changes. These item/supplier/origin country/location combinations are then used to create new cost changes on the date of their tax law changes. The normal cost change processes then take care of updating RMS with the new tax law information when necessary.

The work is done in the L10N\_BR\_T\_EXTAX\_MAINT\_SQL package.

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Phase 1
Pre-Processing	None

Schedule Information	Description
Post-Processing	None
Threading Scheme	Not applicable

### Restart/Recovery

Not applicable

### Key Tables Affected

Tables	Select	Insert	Update	Delete
L10N_BR_EXTAX_REFRESH_CONFIG	Yes	No	No	No
L10N_BR_T_TAX_CALL_RES_FSC_CNT	Yes	No	No	Yes
L10N_BR_EXTAX_REFRESH_COST	Yes	Yes	No	Yes
V_BR_T_STORE	Yes	No	No	No
L10N_BR_ENTITY_CNAE_CODES	Yes	No	No	No
V_BR_T_PARTNER	Yes	No	No	No
V_BR_T_WH	Yes	No	No	No
WH	Yes	No	No	No
V_BR_T_SUPS	Yes	No	No	No
L10N_BR_SUP_TAX_REGIME	Yes	No	No	No
ADDR	Yes	No	No	No
COUNTRY_TAX_JURISDICTION	Yes	No	No	No
COUNTRY	Yes	No	No	No
STATE	Yes	No	No	No
L10N_BR_EXTAX_HELP_NV_PAIR	Yes	Yes	Yes	Yes
L10N_BR_T_EXTAX_DEST_GRP_HELP	Yes	Yes	Yes	Yes
L10N_ATTRIB	Yes	No	No	No
L10N_ATTRIB_GROUP	Yes	No	No	No
EXT_ENTITY	Yes	No	No	No
L10N_BR_T_TAX_CALL_STG_FSC_FDN	Yes	Yes	No	Yes
L10N_BR_T_TAX_CALL_STAGE_ROUTE	Yes	Yes	No	Yes
RETAIL_SERVICE_REPORT_URL	Yes	No	No	No
L10N_BR_T_TAX_CALL_STAGE_RMS	Yes	No	No	Yes

Tables	Select	Insert	Update	Delete
L10N_BR_T_TAX_CALL_RES_TAXCNTR	Yes	No	No	Yes
L10N_BR_T_TAX_CALL_RES_SRVCPRD	Yes	No	No	Yes
L10N_BR_T_TAX_CALL_RES_ECO_CLS	Yes	No	No	Yes
L10N_BR_T_TAX_CALL_RES_ITM_RL	Yes	No	No	Yes
L10N_BR_T_TAX_CALL_RES_ITM_TAX	Yes	No	No	Yes
L10N_BR_T_TAX_CALL_RES_ITM	Yes	No	No	Yes
L10N_BR_T_TAX_CALL_RES	Yes	No	No	Yes
L10N_BR_T_TAX_STAGE_ECO	Yes	No	No	Yes
L10N_BR_T_TAX_STAGE_FIS_ENTITY	Yes	No	No	Yes
L10N_BR_T_TAX_STAGE_ITEM	Yes	No	No	Yes
L10N_BR_T_TAX_STAGE_NAME_VALUE	Yes	No	No	Yes
L10N_BR_T_TAX_STAGE_ORDER	Yes	No	No	Yes
L10N_BR_T_TAX_STAGE_ORDER_EXP	Yes	No	No	Yes
L10N_BR_T_TAX_STAGE_ORDER_INFO	Yes	No	No	Yes
L10N_BR_T_TAX_STAGE_REGIME	Yes	No	No	Yes
L10N_BR_T_TAX_CALL_RES_FSC_FND	Yes	No	No	Yes
L10N_BR_EXTAX_REFRESH_GTT	Yes	Yes	No	Yes
GTAX_COST_CHANGE_GTT	Yes	Yes	Yes	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_COUNTRY_LOC	Yes	No	No	No
COST_SUSP_SUP_HEAD	No	Yes	No	No
COST_SUSP_SUP_DETAIL_LOC	No	Yes	No	No
COST_EVENT_COST_CHG	Yes	Yes	No	No
COST_EVENT	Yes	Yes	No	No
COST_EVENT_RESULT	Yes	No	No	No
COST_EVENT_RUN_TYPE_CONFIG	Yes	No	No	No
L10N_BR_FISCAL_RECLASS_NV	Yes	No	No	No
L10N_BR_FISCAL_RECLASS	Yes	No	No	No
VAT_CODES	Yes	No	No	No

Tables	Select	Insert	Update	Delete
COUNTRY_ATTRIB	Yes	No	No	No
FM_SYSTEM_OPTIONS	Yes	No	No	No
L10N_BR_T_EXTAX_STG_COST	Yes	No	No	No
L10N_BR_T_EXTAX_RES_COST	No	Yes	No	No
V_BR_ITEM_FISCAL_ATTRIB	Yes	No	No	No
ITEM_SUPPLIER	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
COST_EVENT_DEAL	Yes	No	No	No
COST_EVENT_RESULT	No	Yes	No	No

## l10n\_br\_refresh\_extax\_process\_retail

<b>Module Name</b>	l10n_br_refresh_extax_process_retail.ksh
<b>Description</b>	
<b>Functional Area</b>	TaxWeb Integration
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This module picks the staged groups placed in the staging table L10N\_BR\_T\_EXTAX\_STG\_RETAIL by l10n\_br\_refresh\_extax\_setup\_retail.ksh and calls the external tax provider with them. The results of these calls are placed on the staging L10N\_BR\_T\_EXTAX\_RES\_RETAIL and L10N\_BR\_T\_EXTAX\_RES\_RETAIL\_DET tables.

The work is done in the L10N\_BR\_T\_EXTAX\_MAINT\_SQL package.

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Phase 1
Pre-Processing	l10n_br_refresh_extax_setup_retail.ksh
Post-Processing	l10n_br_refresh_extax_finish_retail.ksh
Threading Scheme	N/A

### Restart/Recovery

Not applicable

## Key Tables Affected

Tables	Select	Insert	Update	Delete
RETAIL_SERVICE_REPORT_URL	Yes	No	No	No
L10N_BR_T_TAX_CALL_STAGE_ROUTE	Yes	No	No	No
VAT_CODES	Yes	No	No	No
L10N_BR_T_EXTAX_STG_RETAIL	Yes	No	No	No
L10N_BR_EXTAX_HELP_NV_PAIR	Yes	No	No	No
L10N_BR_T_EXTAX_RES_RETAIL	No	Yes	No	No
L10N_BR_T_EXTAX_RES_RETAIL_DET	No	Yes	No	No

## l10n\_br\_refresh\_extax\_setup\_retail

<b>Module Name</b>	l10n_br_refresh_extax_setup_retail.ksh
<b>Description</b>	
<b>Functional Area</b>	TaxWeb Integration
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This module obtains a list of fiscal attributes that have tax law changes scheduled against them since the last run. It then uses this information to determine what item/locations are affected by those scheduled tax law changes. It determines the unique fiscal attribute groups that cover those item/locations combinations and puts the groups on a stage table (L10N\_BR\_T\_EXTAX\_STG\_RETAIL).

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Phase 1
Pre-Processing	N/A
Post-Processing	l10n_br_refresh_extax_process_retail.ksh
Threading Scheme	N/A

**Restart/Recovery**

N/A

**Key Tables Affected**

<b>Tables</b>	<b>Select</b>	<b>Insert</b>	<b>Update</b>	<b>Delete</b>
L10N_BR_EXTAX_REFRESH_CONFIG	Yes	No	No	No
L10N_BR_T_TAX_CALL_RES_FSC_CNT	Yes	No	No	No
L10N_BR_EXTAX_REFRESH_RETAIL	Yes	Yes	No	Yes
L10N_BR_T_EXTAX_STG_RETAIL	Yes	Yes	No	Yes
L10N_BR_T_EXTAX_RES_RETAIL	No	No	No	Yes
L10N_BR_T_EXTAX_RES_RETAIL_DET	No	No	No	Yes
L10N_BR_T_TAX_CALL_STG_FSC_FDN	Yes	Yes	No	Yes
L10N_BR_T_TAX_CALL_STAGE_ROUTE	Yes	Yes	No	No
RETAIL_SERVICE_REPORT_URL	Yes	No	No	No
L10N_BR_T_TAX_CALL_STAGE_RMS	Yes	No	No	Yes
L10N_BR_T_TAX_CALL_RES_TAXCNTR	Yes	No	No	Yes
L10N_BR_T_TAX_CALL_RES_SRVCPRD	Yes	No	No	Yes
L10N_BR_T_TAX_CALL_RES_ECO_CLS	Yes	No	No	Yes
L10N_BR_T_TAX_CALL_RES_ITM_RL	Yes	No	No	Yes
L10N_BR_T_TAX_CALL_RES_ITM_TAX	Yes	No	No	Yes
L10N_BR_T_TAX_CALL_RES_ITM	Yes	No	No	Yes
L10N_BR_T_TAX_CALL_RES	Yes	No	No	Yes
L10N_BR_T_TAX_STAGE_ECO	Yes	No	No	Yes
L10N_BR_T_TAX_STAGE_FIS_ENTITY	Yes	No	No	Yes
L10N_BR_T_TAX_STAGE_ITEM	Yes	No	No	Yes
L10N_BR_T_TAX_STAGE_NAME_VALUE	Yes	No	No	Yes
L10N_BR_T_TAX_STAGE_ORDER	Yes	No	No	Yes

Tables	Select	Insert	Update	Delete
L10N_BR_T_TAX_STAGE_ORDER_EXP	Yes	No	No	Yes
L10N_BR_T_TAX_STAGE_ORDER_INFO	Yes	No	No	Yes
L10N_BR_T_TAX_STAGE_REGIME	Yes	No	No	Yes
L10N_BR_T_TAX_CALL_RES_FSC_FND	Yes	No	No	Yes
L10N_BR_EXTAX_REFRESH_GTT	Yes	Yes	Yes	Yes
V_BR_ITEM_FISCAL_ATTRIB	Yes	No	No	No
V_BR_T_STORE	Yes	No	No	No
L10N_BR_ENTITY_CNAE_CODES	Yes	No	No	No
V_BR_T_PARTNER	Yes	No	No	No
V_BR_T_WH	Yes	No	No	No
WH	Yes	No	No	No
V_BR_T_SUPS	Yes	No	No	No
L10N_BR_SUP_TAX_REGIME	Yes	No	No	No
ADDR	Yes	No	No	No
COUNTRY_TAX_JURISDICTION	Yes	No	No	No
COUNTRY	Yes	No	No	No
STATE	Yes	No	No	No
L10N_BR_EXTAX_HELP_NV_PAIR	Yes	Yes	Yes	Yes
L10N_BR_T_EXTAX_DEST_GRP_HELP	Yes	Yes	Yes	Yes
ITEM_MASTER	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ITEM_SUPP_COUNTRY_DIM	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
VAT_CODES	Yes	No	No	No
L10N_BR_T_TAX_CALL_RES_FSC_CNT	No	No	No	Yes
L10N_BR_T_TAX_CALL_STAGE_ROUTE	No	No	No	Yes
L10N_ATTRIB	Yes	No	No	No
L10N_ATTRIB_GROUP	Yes	No	No	No
EXT_ENTITY	Yes	No	No	No

## l10nbr\_taxweb\_fisdnl (Taxweb Fiscal Download)

<b>Module Name</b>	l10nbr_taxweb_fisdnl.pc
<b>Description</b>	
<b>Functional Area</b>	TaxWeb Integration
<b>Module Type</b>	
<b>Module Technology</b>	ProC
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This batch loads all fiscal attributes tables in ORFM from the external tax engine. If it is necessary to download or update a specific fiscal attribute, a parameter can be passed as input, but if it needs to download all of them, user can inform ALL as parameter.

For this batch, RTIL is used to retrieve the request to Taxweb Tax Engine to have latest updated fiscal attribute. The cutoff date to download the data from TaxWeb Tax Engine is fetched from field LAST\_UPD\_DATE in FISCAL\_ATTRIB\_UPDATES table. Each time this batch is executed, last execution date is updated in LAST\_UPD\_DATE field.

If any fiscal attributes are returned, they will be loaded (merged) to the corresponding table for that fiscal attribute. The tables loaded per attribute are as follows:

Attribute	Loaded Table
NCM	L10N_BR_NCM_CODES
NCM_CHAR_CODES	L10N_BR_NCM_CHAR_CODES
NCM_IPI_CODES	L10N_BR_NCM_IPI_CODES
NCM_PAUTA_CODES	L10N_BR_NCM_PAUTA_CODES
FEDERAIL_SERVICE_CODES	L10N_BR_FEDERAL_SVC_CODES
MASSERVICE_CODES	L10N_BR_UNIFIED_SVC_CODES
CNAE_CODES	L10N_BR_CNAE_CODES
CEST	L10N_BR_NCM_CEST_CODES
NOP	FM_NOP
ALL	All above tables

---

**Note:** A new attribute (CESTFULL) was created to maintain NCM and CEST codes relationship. When the batch is executed with this parameter, relationship table between NCM and CEST is refreshed when there are not exist in the TaxWeb integration.

---

## Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	This program can run in ad-hoc basis whenever the new fiscal attributes needs to be downloaded from the TaxWeb Tax Engine.
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

## Restart/Recovery

Not applicable

## Key Tables Affected

Tables	Select	Insert	Update	Delete
L10N_BR_NCM_CODES	Yes	Yes	Yes	No
L10N_BR_NCM_CHAR_CODES	Yes	Yes	Yes	No
L10N_BR_NCM_PAUTA_CODES	Yes	Yes	Yes	No
L10N_BR_FEDERAL_SVC_CODES	Yes	Yes	Yes	No
L10N_BR_UNIFIED_SVC_CODES	Yes	Yes	Yes	No
L10N_BR_CNAE_CODES	Yes	Yes	Yes	No
FISCAL_ATTRIB_UPDATES	Yes	Yes	No	No
L10N_BR_T_TAX_CALL_STAGE_FSC_FDN	Yes	Yes	No	Yes
L10N_BR_T_TAX_CALL_STAGE_ROUTE	Yes	Yes	No	Yes
RETAIL_SERVICE_REPORT_URL	Yes	No	No	No
L10N_BR_T_TAX_CALL_RES_FSC_CNT	No	Yes	No	Yes
L10N_BR_T_TAX_CALL_RES_FSC_FND	Yes	Yes	No	Yes
FM_NOP	No	Yes	Yes	No
L10N_BR_NCM_IPI_CODES	Yes	Yes	Yes	No
L10N_BR_NCM_CEST_CODES	Yes	Yes	Yes	No

## Financial Posting Batches

### fmtrandata (FM Transaction Data)

<b>Module Name</b>	fmtrandata.pc
<b>Description</b>	
<b>Functional Area</b>	Financial Postings
<b>Module Type</b>	
<b>Module Technology</b>	ProC
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

#### Design Overview

Computes transactional data based on the transaction codes.

#### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Ad hoc
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	Multi-threaded based on location

#### Restart/Recovery

The logical unit of work is location\_id and fiscal\_doc\_id.

#### Key Tables Affected

Tables	Select	Insert	Update	Delete
FM_TRAN_DATA	No	Yes	No	No
FM_FISCAL_DOC_HEADER	Yes	No	Yes	No
FM_TRAN_CODES	Yes	No	No	No

### fmfinpost (FM Financial Posting)

<b>Module Name</b>	fmfinpost.pc
--------------------	--------------

**Description**

<b>Functional Area</b>	Financial Postings
<b>Module Type</b>	
<b>Module Technology</b>	ProC
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

**Design Overview**

Rolls up the transaction amount into accounts based on general ledger.

**Scheduling Constraints**

<b>Schedule Information</b>	<b>Description</b>
Processing Cycle	Ad hoc
Scheduling Considerations	Ad hoc
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

**Restart/Recovery**

Not applicable

**Key Tables Affected**

<b>Tables</b>	<b>Select</b>	<b>Insert</b>	<b>Update</b>	<b>Delete</b>
FM_GL_OPTIONS	Yes	No	No	Yes
FM_COA_SETUP	Yes	No	No	No
FM_SOB_SETUP	Yes	No	No	No
FM_ACCOUNT_SETUP	Yes	No	No	No
FM_DYNAMIC_SEGMENT_SETUP	Yes	Yes	No	No
FM_GL_CROSS_REF	Yes	Yes	No	No
FM_GL_DYNAMIC_ATTRIBUTES	Yes	Yes	No	No
FM_TRAN_DATA	Yes	No	No	No
FM_FISCAL_DOC_HEADER	Yes	No	Yes	No
FM_SCHEDULE	Yes	No	Yes	No
FM_AP_STAGE_HEAD	Yes	Yes	No	No
FM_AP_STAGE_DETAIL	Yes	Yes	No	No
STG_FIF_GL_DATA	Yes	Yes	No	No

## import\_SPED (SPED Batch)

<b>Module Name</b>	Import_SPED.ksh
<b>Description</b>	
<b>Functional Area</b>	SPED
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This batch is used to post NF processed transactions (status = FP) into SPED staging tables for SPED (SPED - FM\_SPED\_FISCAL\_DOC\_HEADER and FM\_SPED\_FISCAL\_DOC\_DETAIL) reporting by Fiscal Partners.

It looks into the driving table, FM\_SPED\_LAST\_RUN\_DATE, for the last run date of SPED to fetch all closed NFs whose transaction amounts are also rolled up into ledger accounts based on gl\_cross\_ref in between last run date and sysdate from the main tables of ORFM. Once the records are successfully inserted, the batch updates the last\_run\_date date to sysdate.

It is recommended that run this batch job on a daily basis due to performance impacts. This batch has a pre-dependency on the ORFM Financial postings batch, so once this is completed and sets Nota Fiscal status to 'F'inancially 'P'osted (FP).

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	This program should run only after the successful completion of FMFINPOST.PC
Pre-Processing	fmfinpost.pc
Post-Processing	N/A
Threading Scheme	N/A

### Restart/Recovery

Not applicable

### Key Tables Affected

Tables	Select	Insert	Update	Delete
FM_FISCAL_DOC_HEADER	Yes	No	No	No
FM_FISCAL_DOC_DETAIL	Yes	No	No	No

Tables	Select	Insert	Update	Delete
FM_SCHEDULE	Yes	No	No	No
FM_FISCAL_DOC_TAX_HEAD	Yes	No	No	No
FM_FISCAL_DOC_TAX_DETAIL	Yes	No	No	No
FM_FISCAL_DOC_TAX_DETAIL_EXT	Yes	No	No	No
FM_FISCAL_UTILIZATION	Yes	No	No	No
FM_FISCAL_DOC_PAYMENTS	Yes	No	No	No
FM_SPED_FISCAL_DOC_HEADER	No	Yes	No	No
FM_SPED_FISCAL_DOC_DETAIL	No	Yes	No	No
ITEM_MASTER	Yes	No	No	No
FM_SPED_LAST_RUN_DATE	Yes	No	Yes	No
V_BR_ITEM_FISCAL_ATTRIB	Yes	No	No	No
V_FISCAL_ATTRIBUTES	Yes	No	No	No

## Deal Income Postings Batch

Use this batch program to create NFs in Approved status.

### fmdealinc (Deal Income Posting)

<b>Module Name</b>	fmdealinc.pc
<b>Description</b>	
<b>Functional Area</b>	Deal income posting
<b>Module Type</b>	
<b>Module Technology</b>	ProC
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This batch program reads the data from RMS staging tables STAGE\_FIXED\_DEAL\_HEAD, STAGE\_FIXED\_DEAL\_DETAIL, STAGE\_COMPLEX\_DEAL\_HEAD, and STAGE\_COMPLEX\_DEAL\_DETAIL and creates the NFs in Approved status. If the NF creation fails for some fatal reason, the error is logged in the error file. This batch runs on a daily basis after vendinvc and vendinvf RMS batches.

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc

Schedule Information	Description
Scheduling Considerations	Ad hoc
Pre-Processing	vendinvc and vendinvf RMS batches
Post-Processing	N/A
Threading Scheme	Multi-threaded based on deal_id

### Restart/Recovery

The logical unit of work is Deal\_id, Deal\_detail\_id, and location\_id.

### Key Tables Affected

Tables	Select	Insert	Update	Delete
FM_FIXED_DEAL_LOC_DEFAULT	Yes	No	No	No
STAGE_FIXED_DEAL_HEAD	Yes	No	No	No
STAGE_FIXED_DEAL_DETAIL	Yes	No	No	No
STAGE_COMPLEX_DEAL_HEAD	Yes	No	No	No
STAGE_COMPLEX_DEAL_DETAIL	Yes	No	No	No
TSF_ENTITY_ORG_UNIT_SOB	Yes	No	No	No
MV_L10N_ENTITY	Yes	No	No	No
SYSTEM_OPTIONS	Yes	No	No	No
FM_SYSTEM_OPTIONS	Yes	No	No	No
FM_FISCAL_DOC_HEADER	Yes	Yes	No	No
FM_FISCAL_DOC_DETAIL	No	Yes	No	No

## ORFM Utility Batches

### fm\_async\_queue\_cleanup (cleanup queues from async ORFM batches)

<b>Module Name</b>	fm_async_queue_cleanup.ksh
<b>Description</b>	
<b>Functional Area</b>	Queue cleanup
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

## Design Overview

This batch program keeps queues that manages asynchronous ORFM batches cleaned up.

## Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Ad hoc
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

## Restart/Recovery

Not applicable

## Key Tables Affected

Tables	Select	Insert	Update	Delete
RMS_ASYNC_STATUS	Yes	No	Yes	No
RMS_ASYNC_RETRY	Yes	Yes	No	No

## fm\_batch\_avg\_sales\_price (Tax Calculation of Legislation of MG state)

<b>Module Name</b>	fm_batch_avg_sales_price.ksh
<b>Description</b>	
<b>Functional Area</b>	Calculation for average sales price for items
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

## Design Overview

This batch supports new legislation of state Minas Gerais (MG) for new calculation for average sales price for items.

This process should be executed monthly to calculate and store the unit average retail price for domestic sales to final consumers.

In general, this process should address:

- Calculate and store the values of 'Unit Weighted Average Sales Price' per: (Item x Location) and (Item x State) for a specific period;

- Should be scheduled to perform every month to calculate the values for some specific period (month/year). It could be reported (parameter) or deduced automatically.
  - Create a parameter to allow the user to inform the period to be processed:
    - \* For every new execution, if there is data for the period the process should delete all the data first and then start to calculate and store the new data.
    - \* If there is no period reported in the parameter, the schedule will execute the period preceding (mm/aaaa) the execution date.

**Scheduling Constraints**

Schedule Information	Description
Processing Cycle	At the end of RMS batch cycle (Daily).
Scheduling Considerations	This program is scheduled once a month.
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	Location.

**Restart/Recovery**

N/A

**Key Tables Affected**

Tables	Select	Insert	Update	Delete
TRAN_DATA_HISTORY	Yes	No	No	No
FM_FISCAL_DOC_HEADER_HIST	Yes	No	No	No
FM_FISCAL_DOC_DETAIL_HIST	Yes	No	No	No
FM_FISCAL_DOC_HEADER	Yes	No	No	No
FM_FISCAL_DOC_DETAIL	Yes	No	No	No
FM_AVERAGE_SALES_PRICE	Yes	Yes	Yes	No

**fm\_batch\_freclass\_sql (ORFM Reclassification Process)**

<b>Module Name</b>	fm_batch_freeclass_sql.ksh
<b>Description</b>	
<b>Functional Area</b>	Reclassification
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A

<b>Runtime Parameters</b>	N/A
---------------------------	-----

### Design Overview

This batch process fiscal attribute reclassification schedules marked as offline and ready to be processed.

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Ad hoc
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

### Key Tables Affected

Tables	Select	Insert	Update	Delete
FM_FRECLASS_SCHEDULE	Yes	No	Yes	No
FM_FRECLASS_CONFIG	Yes	No	No	No
L10N_ATTRIB	Yes	No	No	No
FM_FRECLASS	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
FM_SYSTEM_OPTIONS	Yes	No	No	No
ITEM_COUNTRY	No	Yes	Yes	No
ITEM_SUPPLIER	No	No	No	No
ITEM_LOC	No	No	No	No
FM_FRECLASS_STG	No	Yes	No	Yes

### fm\_generate\_nf\_stkcount (Stock Count)

<b>Module Name</b>	fm_generate_nf_stkcount.ksh
<b>Description</b>	
<b>Functional Area</b>	NF-e Integration
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

## Design Overview

This module creates a NF for inventory adjustment where there is some quantity discrepancy as result of general inventory process.

## Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Ad hoc
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

## Restart/Recovery

Not applicable

## Key Tables Affected

Tables	Select	Insert	Update	Delete
FM_SYSTEM_OPTIONS	Yes	No	No	No
FM_FISCAL_DOC_HEADER	No	Yes	No	No
FM_FISCAL_DOC_DETAIL	No	Yes	No	No
FM_TRAN_DATA_HISTORY	No	Yes	No	No
FM_FISCAL_UTILIZATION	Yes	No	No	No
FM_UTIL_ADJUST_REASON	Yes	No	No	No
FM_STG_ADJUST_DTL	Yes	No	No	No

## fm\_reprocess\_nf\_status\_f (Reprocess NF-e with status F)

<b>Module Name</b>	fm_reprocess_nf_status_f.ksh
<b>Description</b>	
<b>Functional Area</b>	NF-e Integration
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

## Design Overview

This module reprocesses all NF-e with status F (Post Sefaz Approve Error) to A (Approved).

## Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Ad hoc
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

## Restart/Recovery

Not applicable

## Key Tables Affected

Tables	Select	Insert	Update	Delete
FM_FISCAL_DOC_HEADER	Yes	No	Yes	No
FM_SCHEDULE	Yes	No	Yes	No

## fm\_stock\_position (Stock Position)

Module Name	fm_stock_position.ksh
Description	
Functional Area	
Module Type	
Module Technology	Shell Script
Catalog ID	N/A
Runtime Parameters	N/A

## Design Overview

This module provides information about a snapshot of the stock position in RMS in table FM\_STOCK\_POSITION. This process is asynchronous, and this execution is controlled by parameters.

- Location
  - If executed without parameters, it will take the stock position snap-shot for all locations.
- Data of Fiscal Documents
  - Business tables (default and always used)
  - History tables (optional depend on the parameter ('Y' or 'N')(default))
    - \* If choose 'Y' then Hot and Hist tables will be used;

- \* If choose 'N' then only Hot Tables will be used.

### **Stock Owner (Retailer) - Stock Possession (Retailer)**

The tables ITEM\_MASTER and ITEM\_LOC\_SOH will be main source for the batch process.

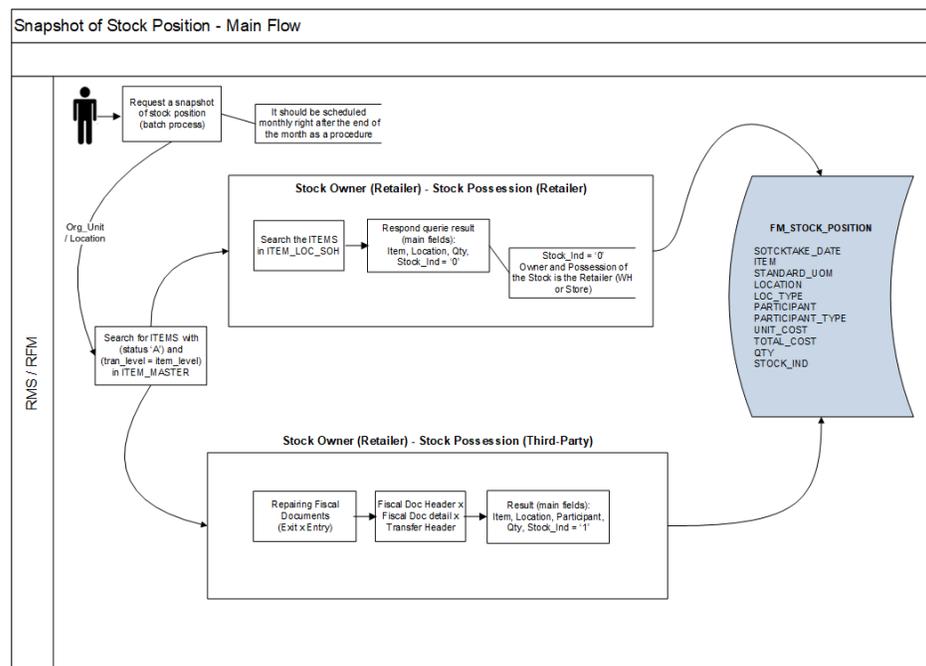
- ITEM\_MASTER
  - status = 'A'
- ITEM\_LOC\_SOH
  - item = item (ITEM\_MASTER)
  - loc = <depending on the parameter informed in the schedule of the process>
  - loc\_type = <depending on the parameter informed in the schedule of the process>

### **Stock Owner (Retailer) - Stock Possession (Third-Party)**

This scenario is about stock in possession of a Third-Party and the stock owner is the Retailer. This search will cover all the active Items for the specific Org Unit / Location. Below is the rule to search and retrieve the data from the source tables:

- ITEM\_MASTER
  - Status = 'A'.
- FM\_FISCAL\_DOC\_HEADER (or HIST) / FM\_FISCAL\_DOC\_DETAIL (or HIST) /
- FM\_FISCAL\_UTILIZATION
  - Requisition Type = 'REP';
  - Status NF: A, C or FP;
  - Utilization mode type = 'EXIT';
  - item = item (ITEM\_MASTER)
  - loc = <depending on the parameter informed in the schedule of the process>
  - loc\_type = <depending on the parameter informed in the schedule of the process>
  - Requisition No.
- TSFHEAD
  - Transfer Parent No = Requisition No (FM\_FISCAL\_DOC\_DETAIL)
- The search must group and total values for the data found using the functional key below.
  - ITEM
  - LOCATION
  - LOCATION\_TYPE
  - PARTICIPANT
  - PARTICIPANT\_TYPE
  - UNIT\_COST
- After grouping following the rule above, the totals should be calculated according to the formula below:

- Quantity of Stock with Third-Party Possession =  
(sum(FM\_FISCAL\_DOC\_DETAIL.QUANTITY for NF Exit Repairing) -  
(sum(FM\_FISCAL\_DOC\_DETAIL.QUANTITY for NF Entry Repairing))
- If there is no NF Entry Repairing related to the NF Exit Repairing, then to apply the formula above, the quantity for NF Entry Repairing should be considered as zero value ('0').
- If the result of the formula above is **greater than zero (>0)**, then a line register should **be inserted** in the table FM\_STOCK\_POSITION. It means that there is a remaining quantity of merchandise stock in possession of the Third-Party.
- If the result of the formula above is **NOT greater than zero (<=0)**, then a line register should **NOT be inserted** in the table FM\_STOCK\_POSITION. It means that there isn't a remaining quantity of merchandise stock in possession of the Third-Party.



### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Ad hoc
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

### Restart/Recovery

Not applicable

## Key Tables Affected

Tables	Select	Insert	Update	Delete
WH	Yes	No	No	No
STORE	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_LOC_SOH	Yes	No	No	No
PACKITEM_BREAKOUT	Yes	No	No	No
FM_SYSTEM_OPTIONS	Yes	No	No	No
FM_FISCAL_DOC_HEADER	Yes	No	No	No
FM_FISCAL_DOC_DETAIL	Yes	No	No	No
TSFHEAD	Yes	No	No	No
FM_STOCK_POSITION	Yes	Yes	No	No

## fmedinf (NF EDI Batch)

<b>Module Name</b>	fmedinf.pc
<b>Description</b>	
<b>Functional Area</b>	NF creation and searching for an item
<b>Module Type</b>	
<b>Module Technology</b>	ProC
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This batch is used to generate NF from EDI staging tables.

This EDI batch runs to create the NF from the entries in the EDI NF tables, in Worksheet status. This NF is not linked to any schedule. Errors can only be viewed in the error log file. A new column in the NF header table displays if the NF has been created manually or through the EDI batch process.

The FM\_EDI\_DOC\_DETAIL table is modified to include a new field, called VPN. This field allows the user to specify the VPN number, if the user is not aware of the RMS item number.

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Ad hoc

Schedule Information	Description
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	Multi-threaded based on Supplier

### Restart/Recovery

The logical unit of work is Supplier and edi\_doc\_id.

### Key Tables Affected

Tables	Select	Insert	Update	Delete
FM_EDI_DOC_HEADER	Yes	No	Yes	No
FM_EDI_DOC_DETAIL	Yes	No	No	No
FM_EDI_DOC_PAYMENT	Yes	No	No	No
FM_EDI_DOC_COMPLEMENT	Yes	No	No	No
FM_EDI_IMPORT_HEADER	Yes	No	No	No
FM_EDI_IMPORT_DETAIL	Yes	No	No	No
FM_FISCAL_IMPORT_HEADER	Yes	Yes	No	No
FM_FISCAL_IMPORT_DETAIL	Yes	Yes	No	No
FM_EDI_DOC_TAX_HEAD	Yes	No	No	No
FM_EDI_DOC_TAX_DETAIL	Yes	No	No	No
FM_FISCAL_DOC_HEADER	Yes	Yes	No	No
FM_FISCAL_DOC_DETAIL	Yes	Yes	No	No
FM_FISCAL_DOC_COMPLEMENT	Yes	Yes	No	No
FM_FISCAL_DOC_PAYMENTS	Yes	Yes	No	No
FM_FISCAL_DOC_TAX_HEAD	No	Yes	No	No
FM_FISCAL_DOC_TAX_DETAIL	Yes	Yes	No	No
FM_UTILIZATION_ATTRIBUTES	Yes	No	No	No
FM_FISCAL_DOC_TAX_HEAD_EXT	Yes	Yes	No	No
FM_FISCAL_DOC_TAX_DETAIL_EXT	Yes	Yes	No	No
FM_ERROR_LOG	No	Yes	No	Yes

### fmmanifest (FM Manifest Batch)

Module Name	fmmanifest.ksh
Description	
Functional Area	Fiscal Receiving

---

**Module Type**

<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

---

**Design Overview**

This batch inserts in the FM\_STG\_NFE the record that indicates to the Partner DFe the "Manifestação do Destinatário SEFAZ" request for the "Operação Não Realizada" (status = NFE\_C\_P) and "Desconhecimento da Operação" (status = NFE\_U\_P) events.

Are eligible for the event "Operação Não Realizada": the fiscal documents that are in the FM\_FISCAL\_DOC\_HEADER table inactive (status = I), and the fiscal document previews that are in the FM\_IRL\_STATUS table refused (status: 305- REFUSED BY DISAGREEMENT, 306 - REFUSED AUTOMATICALLY AFTER PRE-VALIDATION, 6- REFUSED RECEIVING or 802- PHYSICAL RECEIVING REFUSED).

Are eligible for the manifestation of the event "Desconhecimento da Operação" are the fiscal document previews that are in the FM\_IRL\_STATUS table refused exclusively by unknown (status: 304- REFUSED BY UNKNOWN OPERATION).

In addition to the eligibility status criterion, the batch considers the number of days defined through the MDEST\_SEFAZ System Option. Only records that have status definition dates less than or equal to the VDATE added to the defined number of days are processed.

If the record has been successfully inserted into the FM\_STG\_NFE table, the batch changes the status of the fiscal document (status = NFE\_MP), and the status of the fiscal document previews (status: 1001- WAITING SEFAZ CONFIRMATION) to Manifest Pending.

Single and daily execution of this batch is recommended.

**Scheduling Constraints**

<b>Schedule Information</b>	<b>Description</b>
Processing Cycle	Daily
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

---

**Restart/Recovery**

Not applicable

**Key Tables Affected**

<b>Tables</b>	<b>Select</b>	<b>Insert</b>	<b>Update</b>	<b>Delete</b>
FM_FISCAL_DOC_HEADER	Yes	No	Yes	No

Tables	Select	Insert	Update	Delete
FM_IRL_STATUS	Yes	No	Yes	No
FM_IRL_HEADER	Yes	No	No	No
FM_IRL_LOG	No	Yes	No	No
FM_NFE_STG	Yes	Yes	No	No
FM_UTILIZATION_ATTRIBUTES	Yes	No	No	No
FM_SYSTEM_OPTIONS	Yes	No	No	No

## fmpprepost (Pre/Post processing)

<b>Module Name</b>	fmpprepost.pc
<b>Description</b>	
<b>Functional Area</b>	Pre/Post functionality for ORFM
<b>Module Type</b>	
<b>Module Technology</b>	ProC
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This batch program has pre/post module for fntaxupld batch. It contains fntaxupld\_post function. This function processes the records from staging table FM\_STG\_TAXCHG\_DTL, contains process\_ind as 'N', and inserts the records into FM\_TAXCHG\_HEADER and FM\_TAXCHG\_DETAIL tables.

This program considers the following four parameters:

- Username/password to log on to Oracle
- A program before or after which this script must run
- An indicator indicating whether the script is a pre or post function
- chunk\_size for fntaxupld only, which defines the number of records under one location that need to be clubbed as single chunk

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Ad hoc
Pre-Processing	fmntaxupld
Post-Processing	N/A
Threading Scheme	N/A

**Restart/Recovery**

Not applicable

**Key Tables Affected**

Table	Select	Insert	Update	Delete
FM_STG_TAXCHG_DTL	Yes	No	Yes	Yes
FM_TAXCHG_HEADER	Yes	Yes	No	No
FM_TAXCHG_DETAIL	Yes	Yes	No	No
V_FISCAL_ATTRIBUTES	Yes	No	No	No
FM_TAXCHG_HEADER_HIST	Yes	No	No	No
FM_TAXCHG_DETAIL_HIST	Yes	No	No	No

**l10n\_id\_iindfiles (Load data into template tables for item induction)**

<b>Module Name</b>	l10n_id_iindfiles.ksh
<b>Description</b>	
<b>Functional Area</b>	Item Induction
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

**Design Overview**

This batch program is process Item Induction files to insert item in system.

**Scheduling Constraints**

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Ad hoc
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

**Restart/Recovery**

Not applicable

## Key Tables Affected

Tables	Select	Insert	Update	Delete
FM_CLEAR_CALC_STG_TABLES	Yes	No	No	Yes

## Localization Purge Batch Process

### fm\_batch\_clear\_calc\_stg\_t (Purge localization stage tables)

<b>Module Name</b>	fm_batch_clear_calc_stg_t.ksh
<b>Description</b>	
<b>Functional Area</b>	Purge
<b>Module Type</b>	
<b>Module Technology</b>	Shell Script
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This batch is used to clear records from the stage tables.

New batch process to get all data informed in table FM\_CLEAR\_CALC\_STG\_TABLES and for each line call function.

- FM\_T\_EXT\_TAXES\_SQL
- CLEAR\_STG\_TABLES

After call the function delete the processed lines in table FM\_CLEAR\_CALC\_STG\_TABLES.

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Ad hoc
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

### Restart/Recovery

This batch program uses table-based restart/recovery. The commit happens in the database when the commit\_max\_ctr is reached.

### Key Tables Affected

Tables	Select	Insert	Update	Delete
FM_CLEAR_CALC_STG_TABLES	Yes	No	No	Yes

### fmpurge (Purging Process Batch)

<b>Module Name</b>	fmpurge.pc
<b>Description</b>	
<b>Functional Area</b>	Purging Process
<b>Module Type</b>	
<b>Module Technology</b>	ProC
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This batch is used to purge the NF records from the transaction and history tables.

In production environment, as the number of transactions increases over a period of time; in order to keep the performance intact it is required to keep purging the data from the active tables of the application periodically. This batch purges the data from the active tables and stores them in history tables.

This batch should be scheduled to purge the data from history tables over a period of time.

### Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Ad hoc
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

### Restart/Recovery

Not applicable

## Key Tables Affected

Table	Select	Insert	Update	Delete
FM_SCHEDULE_HIST	No	Yes	No	No
FM_FISCAL_DOC_HEADER_HIST	No	Yes	No	No
FM_FISCAL_DOC_DETAIL_HIST	No	Yes	No	No
FM_TRAN_DATA_HIST	No	Yes	No	No
FM_RECEIVING_HEADER_HIST	No	Yes	No	No
FM_RECEIVING_DETAIL_HIST	No	Yes	No	No
FM_FISCAL_DOC_TAX_HEAD_HIST	No	Yes	No	No
FM_FISCAL_DOC_TAX_DETAIL_HIST	No	Yes	No	No
FM_RESOLUTION_HIST	No	Yes	No	No
FM_CORRECTION_TAX_DOC_HIST	No	Yes	No	No
FM_CORRECTION_DOC_HIST	No	Yes	No	No
FM_NF_DOC_TAX_HEAD_EXT_HIST	No	Yes	No	No
FM_FISCAL_IMPORT_DETAIL_HIST	No	Yes	No	No
FM_FISCAL_IMPORT_HEADER_HIST	No	Yes	No	No
FM_NF_DOC_TAX_DETAIL_EXT_HIST	No	Yes	No	No
FM_NF_DOC_TAX_DETAIL_WAC_HIST	No	Yes	No	No
FM_FISCAL_DOC_PAYMENTS_HIST	No	Yes	No	No
FM_NF_DOC_TAX_RULE_EXT_HIST	No	Yes	No	No
FM_SPED_FISCAL_DOC_HEADER_HIST	No	Yes	No	No
FM_SPED_FISCAL_DOC_DETAIL_HIST	No	Yes	No	No
FM_SCHEDULE	Yes	No	No	Yes
FM_FISCAL_DOC_HEADER	Yes	No	No	Yes
FM_FISCAL_DOC_DETAIL	Yes	No	No	Yes
FM_TRAN_DATA	Yes	No	No	Yes
FM_RECEIVING_HEADER	Yes	No	No	Yes
FM_RECEIVING_DETAIL	Yes	No	No	Yes
FM_FISCAL_DOC_TAX_HEAD	Yes	No	No	Yes
FM_FISCAL_DOC_TAX_DETAIL	Yes	No	No	Yes

Table	Select	Insert	Update	Delete
FM_RESOLUTION	Yes	No	No	Yes
FM_CORRECTION_TAX_DOC	Yes	No	No	Yes
FM_CORRECTION_DOC	Yes	No	No	Yes
FM_FISCAL_DOC_TAX_HEAD_EXT	Yes	No	No	Yes
FM_FISCAL_DOC_TAX_DETAIL_EXT	Yes	No	No	Yes
FM_FISCAL_DOC_TAX_DEATIL_WAC	Yes	No	No	Yes
FM_FISCAL_DOC_PAYMENTS	Yes	No	No	Yes
FM_FISCAL_DOC_TAX_RULE_EXT	Yes	No	No	Yes
FM_SPED_FISCAL_DOC_HEADER	Yes	No	No	Yes
FM_SPED_FISCAL_DOC_DETAIL	Yes	No	No	Yes
FM_FISCAL_ENTITY_OTHER	Yes	No	No	Yes
FM_FISCAL_ENTITY_OTHER_HIST	Yes	Yes	No	Yes
FM_EDI_FISCAL_ENTITY_OTHER	Yes	No	No	Yes
FM_NF_PURGE_DAYS_SETUP	Yes	No	No	Yes
FM_FRECLASS_STG	Yes	No	No	Yes
FM_FRECLASS	Yes	No	No	Yes
FM_FRECLASS_SCHEDULE	Yes	No	No	Yes

## I10nbrfreclsprg (Fiscal Reclassification Purge)

<b>Module Name</b>	I10nbrfreclsprg.pc
<b>Description</b>	
<b>Functional Area</b>	TaxWeb Integration
<b>Module Type</b>	
<b>Module Technology</b>	ProC
<b>Catalog ID</b>	N/A
<b>Runtime Parameters</b>	N/A

### Design Overview

This batch purges the processed reclassification data from L10N\_BR\_FISCAL\_RECLASS table. The records to be purged are based on its processed\_date or active\_date less than the current vdate along with the status.

## Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	This program can run after the successful completion of the following scripts: <ul style="list-style-type: none"> <li>▪ l10n_br_fiscal_reclass_item_setup_retail.ksh</li> <li>▪ l10n_br_fiscal_reclass_item_process_retail.ksh</li> <li>▪ l10n_br_fiscal_reclass_item_finish_retail.ksh</li> </ul>
Pre-Processing	<ul style="list-style-type: none"> <li>▪ l10n_br_fiscal_reclass_item_setup_retail.ksh</li> <li>▪ l10n_br_fiscal_reclass_item_process_retail.ksh</li> <li>▪ l10n_br_fiscal_reclass_item_finish_retail.ksh</li> </ul>
Post-Processing	N/A
Threading Scheme	Threading based on reclassification ID

## Restart/Recovery

Not applicable

## Key Tables Affected

Table	Select	Insert	Update	Delete
L10N_BR_FISCAL_RECLASS	Yes	No	No	Yes
PERIOD	Yes	No	No	No
L10N_BR_FISCAL_RECLASS_NV	Yes	No	No	Yes

## l10npurge (Localization Purging)

Module Name	l10npurge.pc
Description	
Functional Area	Purging Process
Module Type	
Module Technology	ProC
Catalog ID	N/A
Runtime Parameters	N/A

## Design Overview

This batch is used to purge records from the transaction and history tables.

In production environment, as the number of transactions increases over a period of time; in order to keep the performance intact it is required to keep purging the data from the active tables of the application periodically. This batch purges the data from the active tables and stores them in history tables.

## Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad hoc
Scheduling Considerations	Ad hoc
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

## Restart/Recovery

Not applicable

## Key Tables Affected

Table	Select	Insert	Update	Delete
GTAX_ITEM_ROLLUP	Yes	No	No	Yes
POS_MODS_TAX_INFO	Yes	No	No	Yes
GTAX_ITEM_ROLLUP_HIST	No	Yes	No	No
POS_MODS_TAX_INFO_HIST	No	Yes	No	No

---

## Configuring Security for Web Services

ORFM integrates with a TaxWeb Tax Rules, through the Retail Tax Integration Layer (RTIL) to take advantage of situations that may arise as a result of the Brazil tax legislations.

The communication between ORFM/RMS, the RTIL, and TaxWeb Tax Rules occurs through the Web Services. The Oracle Software Security Assurance (OSSA) program advocates that the Web Service communication should occur in a secure manner.

The different mechanisms of securing web services are:

- Transport-level security, in which Secure Sockets Layer (SSL) is used to secure the connection between a client application and the Web Service.
- Message-level security, in which data in SOAP message is digitally signed or encrypted.
- Access control security, which specifies which users, groups, and roles are allowed to access Web Services.

OSSA mandates that sensitive data like passwords should be encrypted before storing into the database. Encryption/decryption is the approach adopted for securing data where the original data must be reconstructed from the encrypted data. The data is encrypted and decrypted using a key value and a specific algorithm.

The following functional processes in ORFM/RMS use the tax calculation in which integration is realized through secured Web Services:

- Inbound Nota Fiscal validation
- Outbound Nota Fiscal issuing
- PO tax break up
- Item creation (Purchase/Sales)
- Freight Nota Fiscal

This chapter covers the following topics:

- [Transport-Level Security](#)
- [Message-Level Security](#)
- [Access Control Security](#)
- [Implementing Web Service Security](#)

## Transport-Level Security

Transport-level security secures the connection between the client application and WebLogic Server with SSL. SSL provides secure connections by allowing two applications connecting over a network to authenticate the other's identity and by encrypting the data exchanged between the applications. Authentication allows a server, and optionally a client, to verify the identity of the application on the other end of a network connection. Encryption makes data transmitted over the network intelligible only to the intended recipient.

## Message-Level Security

Message-level security includes all the security benefits of SSL, but with additional flexibility and features. Message-level security is end-to-end, which means that a SOAP message is secure even when the transmission involves one or more intermediaries. The SOAP message itself is digitally signed and encrypted, rather than just the connection. And finally, it is possible to specify that only parts of the message be signed or encrypted.

## Access Control Security

For access control security, specify the list of users, groups, or roles that are allowed to access a Web Service. Then, when a client application attempts to invoke a Web Service operation, the client authenticates itself to WebLogic Server, using HTTP, and if the client has the authorization, it is allowed to continue with the invocation. Access control security secures only WebLogic Server resources. This means that it is possible to configure only access control security, the connection between the client application and WebLogic Server is not secure and the SOAP message is in plain text.

## Implementing Web Service Security

The user credentials you can use to invoke the Web Service from Oracle Retail applications (ORFM/RMS) are stored in an encrypted format. The user name and password which is provided at the client side is encrypted by using a database procedure with an algorithm initially and stored into the database. At the time of invocation another database procedure is used to decrypt the password and set the authentication parameters internally in the SOAP message. On the server side, the Web Service custom security policy (usernetoken) is attached to the Web Service method. The same user name and password is attached to the usernetoken policy on the server side.

The user name profile tokens are the user name and password associated with a Web Service request and used for authentication and authorization.

## Testing Web Service Security

Use the following procedure to test your Web Service security:

1. Configure the user name/password at the server side (WebLogic) and attach the user name/password to the username policy.
2. Using the encrypt package, store different encrypted username/password to the database (RETAIL\_SERVICE\_REPORT\_URL table). The Web Service invocation fails if a wrong password is set up for the invocation.

---

---

## Oracle Retail Invoice Receiving Layer

Currently the ORFM provides an interface for automatic generation of the fiscal document, the RFM/EDI, but it does not have a mechanism that generates the fiscal document based directly on the NF-e XML file. To use this interface the customer has to customize process and transform the data from XML to the expected data by the RFM/EDI, considering all the complexity of the rules of deduction and transformation.

To simplify this process, Oracle Retail Receiving Layer (IRL) was conceived to enable the generation of fiscal document in RFM directly from the NF-e XML file. This process provides the capture (when using IRL OSB integration), processing and fiscal validation of the NF-e, transforming it in the expected data for EDI. It supports the entire cycle of processing the NF-e in the RFM, reporting any problems that may occur and through the reporting of progress status of the receiving.

Valid NF-e document models for this process are:

- 55 - Nota Fiscal Eletrônica
- 57 - Conhecimento de Transporte Eletrônico

The following topics are covered in this chapter:

- [IRL Architecture](#)
- [Integration Overview](#)
- [IRL Web Services](#)
- [IRL Data Model](#)

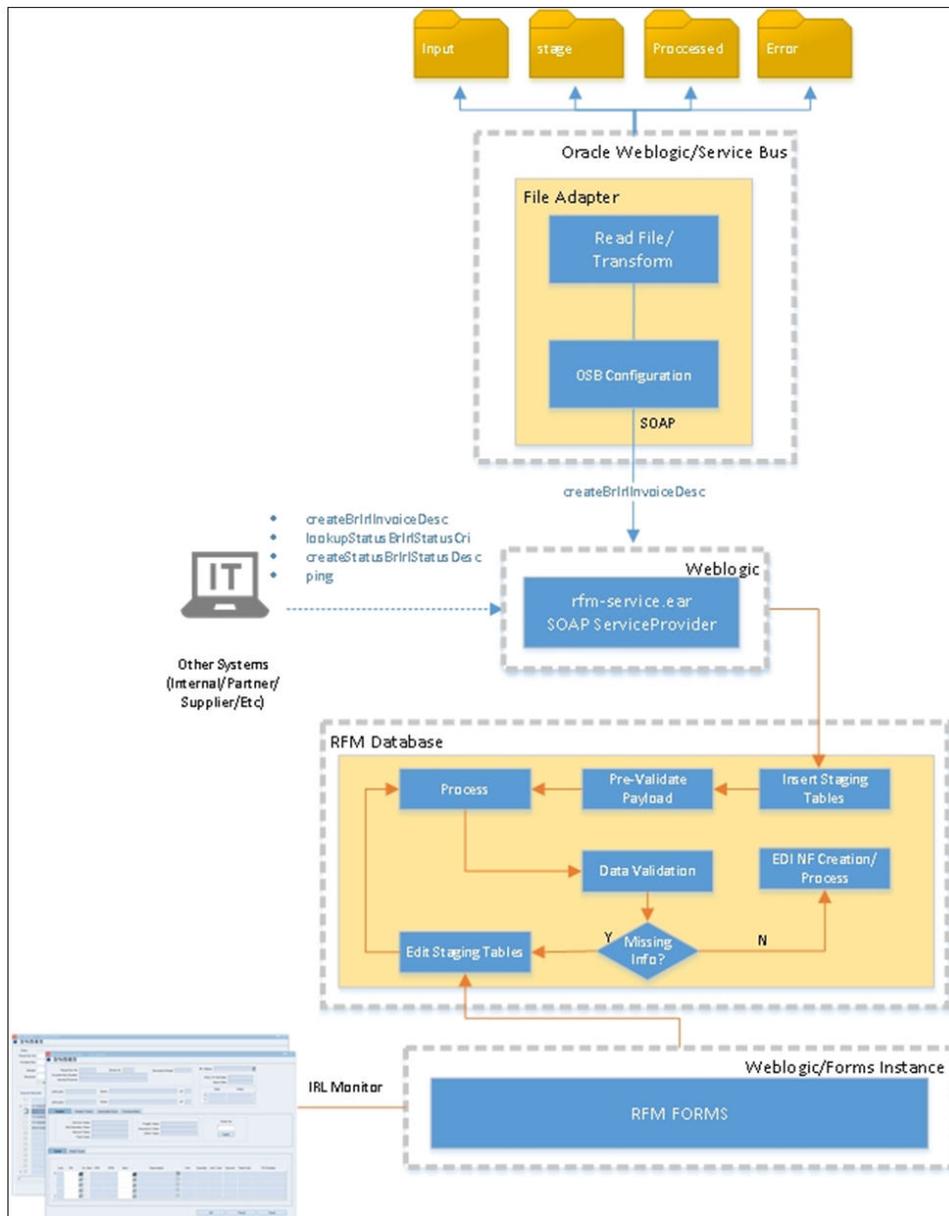
### IRL Architecture

The architecture has been designed in three distinct layers with focus in performance, extensibility, and flexibility:

1. **File capture and transformation:** Using OSB and staging directories this process reads the NF-e XML file, transforms it in IRL data structure and calls the Invoice creations service. This process does not capture the XML file directly from SEFAZ
2. **Data receiving Services (rfm-service):** Using RSE (Retail SOA Enabler) a set of services are exposed to receive the NF-e XML data in a simplified structure. There is also a service to provide the status of the receiving process.
3. **Data validation and Fiscal Document creation:** PL/SQL packages were created to validate the received information and create the NF in RFM (through EDI process). If the process was not able to deduce data (item id for example) a set of forms were created to enable data edition, revalidation, and refusal.

# Integration Overview

Figure 13–1 IRL Architecture Overview



## IRL Web Services

A set of services are available to receive the NF-e data in a simplified structure and provide status of the receiving. The following services are available through rfm-service.ear:

- **createBrIrlInvoiceDesc:** This service validates the received data, deduce values based on system configuration, inserts the validated data into RFM EDI tables and creates the Nota Fiscal in ORFM.
- **lookupStatusBrIrlStatusCri:** This service provides the status of the NF processing.

- createStatusBrIrlStatusDesc:** This service creates a status in IRL tables. It's used by OSB process to identify NF's that are not successfully validated during file capture process.

---

**Note:** To deploy the rfm-service.ear, see the *Oracle Retail Fiscal Management Installation Guide*.

---

## IRL Data Model

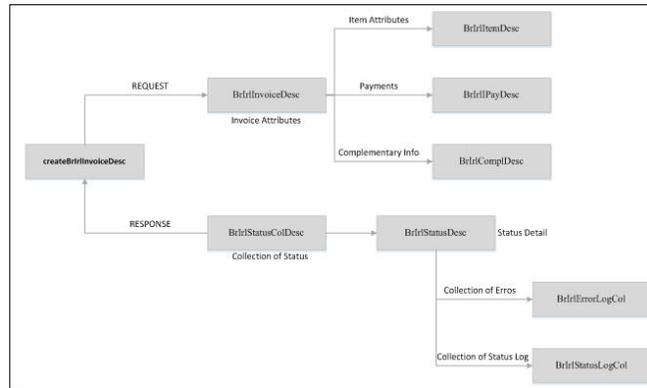
The data model for IRL was designed to simplify the transformation of NF-e XML data into RFM data. It's not necessary to search for id's in RFM in order to create the NF, instead of that only a structural transformation is necessary from NF-e model to IRL model.

---

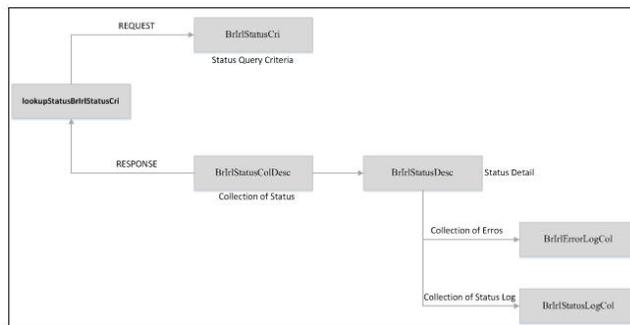
**Note:** IRL OSB component already contains this mapping.

---

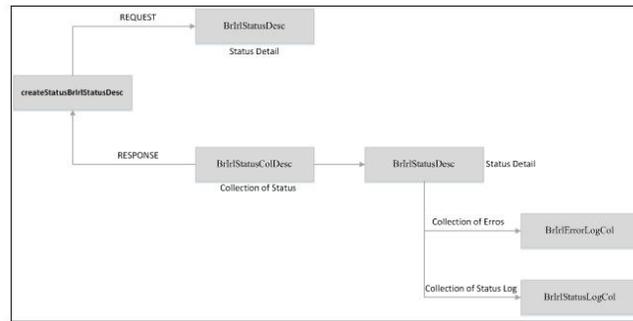
**Figure 13–2 Data Model for createBrIrlInvoiceDesc Service**



**Figure 13–3 Data Model for lookupStatusBrIrlStatusCri Service**



**Figure 13–4 Data Model for createStatusBrIrlStatusDesc Service**



## Object Structure Overview

The object structure overview are as follows:

- **BrIrlInvoiceDesc:** This object represents the header of the fiscal document with main attributes such as document model, number, series, and so on.
- **BrIrlItemDesc:** This object represents the details of a fiscal document. It includes the taxes at item level.
- **BrIrlIPayDesc:** This object represents the list of payments of a fiscal document.
- **BrIrlCompIDesc:** This object represents a list of complementary fiscal document information.
- **BrIrlErrorLog:** This object represents the details of an error found during IRL processing.
- **BrIrlErrorLogCol:** This object is a list of BrIrlErrorLog objects.
- **BrIrlStatusDesc:** This object represents the details of the status of IRL processing for a giving document.
- **BrIrlStatusColDesc:** This object is a collection is BrIrlStatusDesc objects.
- **BrIrlStatusCri:** This object represents the criteria for finding IRL documents.
- **BrIrlStatusLog:** This object represents the status log of IRL process. Every time a status is updated, the previous status is recorded for log tracking.
- **BrIrlStatusLogCol:** This object is a collection of BrIrlStatusLogCol.

## Configuration

Before using the IRL integration some mandatory Fiscal Management System Options should be set. Ensure they are all set before using the integration.

For information about all ORFM system options to setup IRL system behavior, see *Oracle Retail Fiscal Management User Guide*.

---

---

## Tax Burden Display at POS and Sales NF

This chapter covers the following topics:

- [Overview](#)
- [Tax Burden Induction Framework](#)
- [Manual Upload](#)
- [Template Maintenance](#)

### Overview

In Brazil, new legislation related to the consumer protection code was released, determining that the tax burden included in products and services sold to end consumer must be highlighted in the fiscal documents or fiscal tickets in a specific field. The purpose of this law is to provide transparency to consumers on the amount of taxes included in retail prices.

In that way, the law determines that the total amount of federal, state and municipal taxes levied on the sale of goods or services need to be stated in the fiscal documents. Companies must inform "approximate amount of taxes" per item or service sold in a specific field. Also, the header of the fiscal document should include the total amount of taxes, considering all items and services sold.

As the rate cannot be calculated or determined by calculations in the system, it must be uploaded from an external source. It will be necessary to have the average tax rate coming from authorized entities, approved to provide such information. One of the authorized entities that publish the average tax rate for all types of products is the IBPT - Brazilian Institute of Tax Planning.

IBPT provides via downloadable file, the rates for the determination of the approximate values of taxes to the consumer, making it possible for businesses issuing fiscal documents or POS fiscal tickets the full compliance with the law, exempting them from any responsibility for the calculation of this tax, since the source is acknowledged in the documents.

### Tax Burden Induction Framework

In order to support the use case presented above, the Tax Burden Induction framework was created to provide the flexibility to upload data through an open document spreadsheet (".ods" extension) formatted in a predefined template.

## Manual Upload

An upload from a spreadsheet is initiated by user request through the RFM Tax Burden Induction screen. The request includes the source file location and the country state that the data represents. On submission of the upload request, RFM will process the request and perform relevant validation and will import the data into destination tables.

## Template Maintenance

The authorized entities that provide the tax burden data may change the data layout throughout the time. If that happens there are two available options in order to upload data successfully into ORFM.

1. Modify the data spreadsheet to be compliance with previous template, already available in database.
2. Modify the process template (l10n\_template\_config.ods file).

The below section describes how to modify the template document. The template is found in "Localization/s9t\_templates/ l10n\_template\_config.ods".

---



---

**Note:** ORFM does provide a script that can be used to upload template configurations from a spreadsheet. For more information on this function, see *Oracle Retail Merchandising System with Brazil Localization Installation Guide*.

---



---

## Template Header

The template header (S9T\_TEMPLATE sheet) table captures the template ID, description, as well as, a template type that aids in the classification of templates.

The key fields in the header are described as follows:

**Table 14–1 Key Fields in the Template Header**

Configuration Table Column	Description
TEMPLATE_KEY	This is a unique identifier of a template in RMS and should be unique for each new template added.  Do not modify this column. Valid value is only FM_APPROX_TAX_BURDEN_SOURCE for this process.
TEMPLATE_NAME	This column contains the name of the template and will be the displayed name available in the Tax Burden Induction screen when requesting an upload of data.
TEMPLATE_DESC	Description of the template.
FILE_ID	Do not fill this column.
TEMPLATE_TYPE	It indicates if IRL Monitor allows multisite treatment.  Do not modify this column. Valid value is only L10NT for this process.

## Worksheet Definition

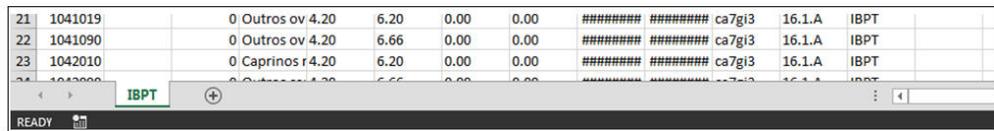
The worksheet definition table lists all tables that should be considered for inclusion as part of a given template. For Tax Burden Induction process only FM\_APPROX\_TAX\_BURDEN\_SOURCE table is available.

The key fields in the header are described as follows:

**Table 14–2 Worksheet Definition Tables**

Configuration Table Column	Description
WKSHT_KEY	Do not modify this column. Valid value is only FM_TAX_RATES.
WKSHT_NAME	This contains the user-facing name that will appear on the tab in a spreadsheet workbook for this table. The description also supports translation for cases where users have different primary languages. It should be noted that the worksheet name is also used to map the worksheets in the spreadsheet to the tables in the staging area when uploading data. Therefore, it should not be changed by users. Valid value example IBPT (see figure Fig x - worksheet name example).
MANDATORY	The table should be designated as mandatory to display it in the spreadsheet generated by this template, or to indicate that it will be present in the upload. Do not modify this column.

**Figure 14–1 Worksheet Name Example**



**Note:** Do not use spaces if Microsoft Excel will be used to modify worksheets or worksheet names.

## Column Definition

The column definition table will contain information on the columns to be included in the template for each table defined as part of worksheet definition. Just as the tables included in a template get mapped to worksheets in the spreadsheet, columns indicated as mandatory in the column definition table are mapped to the columns in the worksheet corresponding to that table.

The key fields in the column definition table are described as follows:

**Table 14–3 Column Definition Table**

<b>Configuration Table Column</b>	<b>Description</b>
COLUMN_KEY	<p>This should be the database name of the column in the impacted table. The columns used for Tax Burden process are (all columns should be in the template):</p> <ul style="list-style-type: none"> <li>■ NCM_NBS_CODE - Classification code</li> <li>■ EX_IPI - Extension of Classification code</li> <li>■ ITEM_TYPE - Item type</li> <li>■ DESCRIPTION - Description of Classification</li> <li>■ TAX_RATE_DOMESTIC_FEDERAL - Domestic Federal Tax Rate</li> <li>■ TAX_RATE_IMPORTED_FEDERAL - Imported Federal Tax Rate</li> <li>■ TAX_RATE_STATE - State Rate</li> <li>■ TAX_RATE_MUNICIPAL - Municipal Rate</li> <li>■ BEGIN_DATE - Begin Date of the Rate</li> <li>■ END_DATE - End Date of the Rate</li> <li>■ KEY_CODE - Code of the spreadsheet (usually provided by the authorized entity)</li> <li>■ VERSION_CODE - Version of the spreadsheet (usually provided by the authorized entity)</li> <li>■ SOURCE_CODE - Source of the information (example, IBPT)</li> <li>■ ACTION - Action to be taken by the process (valid values are NEW to insert the record, MOD to modify an existing record or DEL to delete a record). This column is not mandatory and if it is not informed default value will be "NEW".</li> </ul>
COLUMN_NAME	<p>This contains the name that to display in the spreadsheet column header for this template; for example, código. The description will also support translation for cases where users have different primary languages.</p> <p>Like the worksheet name, the column name is also used to map the worksheet columns to the columns in the table. It is assumed that the column names are not edited by the business users while creating or updating data.</p>
MANDATORY	The column should be designated as mandatory to indicate that the column is mandatory to be present in the upload.
DEFAULT	This contains a default value for the column in case there is no value informed in the column during upload.
REQUIRED_VISUAL_IND	Not used in this process. Leave it blank.

## Spreadsheet Formatting

Given that the spreadsheet is intended to ease the process of data entry and enrichment, certain usability features have been incorporated in the downloaded spreadsheet. However, because the use of spreadsheets to maintain data is so flexible, there are opportunities for users to make changes that may render the data invalid from a Tax Burden Induction perspective. Hence, some key assumptions have been made that dictate how the Tax Burden induction framework processes the data.

The columns marked as mandatory in the column definition tables will form the columns in the respective worksheets. The order in which columns appear in a sheet is not defined as part of template configuration, but users can rearrange columns in an individual sheet for ease of use. This will not impact data upload.

The templates are not specific to the action being taken for the data, which means that users can create, update, and delete information all in a single spreadsheet. The action to be performed with the data on each row (Create, Update, or Delete) will be indicated on the first column of each row on the spreadsheet for data uploads.

**Figure 14–2 Sample Spreadsheet Format**

1	Action	codigo	ex	tipo	descricao nacional	feimportad	estadual	municipal	vigenciai	n	vigenciafi	chave	versao	fonte
2	NEW	1012100			0 Cavalos re	4.20	6.20	18.00	0.00	#####	#####	ca7gi3	16.1.A	IBPT
3	NEW	1012900			0 Cavalos vi	4.20	6.66	18.00	0.00	#####	#####	ca7gi3	16.1.A	IBPT
4	NEW	1013000			0 Asininos	4.20	7.11	18.00	0.00	#####	#####	ca7gi3	16.1.A	IBPT
5	MOD	1019000			0 Outros asi	4.20	7.11	18.00	0.00	#####	#####	ca7gi3	16.1.A	IBPT
6	DEL	1022110			0 Bovinos r	4.20	6.20	0.00	0.00	#####	#####	ca7gi3	16.1.A	IBPT
7		1022190			0 Outros bo	4.20	6.20	0.00	0.00	#####	#####	ca7gi3	16.1.A	IBPT

## Template Definition Examples

See the following sections for template definition examples.

### IBPT Source Template Example

This section illustrates the setup of a simple template Tax Burden Induction process.

The worksheet definition table has records for each of the three tables to be included with the mandatory flag set and the sequence of ordering of worksheets specified by the sequence number column.

**Table 14–4 S9T\_TEMPLATE**

TEMPLATE_KEY	TEMPLATE_NAME	TEMPLATE_DESC	FILE_ID	TEMPLATE_TYPE
FM_APPROX_	Tax Burden	Tax Burden		L10NT
TAX_BURDEN_	Information	Information		
SOURCE				

**Table 14–5 S9T\_TMPL\_WKSHT\_DEF**

TEMPLATE_KEY	WKSHT_KEY	WKSHT_NAME	MANDATORY
FM_APPROX_	FM_TAX_RATES	IBPT	Y
TAX_BURDEN_			
SOURCE			

**Table 14–6 S9T\_TMPL\_COLS\_DEF**

TEMPLATE_KEY	WKSHT_KEY	COLUMN_KEY	COLUMN_NAME	MANDATORY	DEFAULT	REQUIRED_VISUAL_IND
FM_APPROX_	FM_TAX_	NCM_	codigo	Y		Y
TAX_BURDEN_	RATES	NBS_				
SOURCE		CODE				

**Table 14-6 (Cont.) S9T\_TMPL\_COLS\_DEF**

TEMPLATE_ KEY	WKSHT_ KEY	COLUMN_ KEY	COLUMN _NAME	MANDATORY	DEFAULT	REQUIRED_ VISUAL_IND
FM_ APPROX_ TAX_ BURDEN_ SOURCE	FM_ TAX_ RATES	EX_IPI	ex	N		
FM_ APPROX_ TAX_ BURDEN_ SOURCE	FM_ TAX_ RATES	ITEM_ TYPE	tipo	N		
FM_ APPROX_ TAX_ BURDEN_ SOURCE	FM_ TAX_ RATES	DESCRIPT ION	descricao	N		
FM_ APPROX_ TAX_ BURDEN_ SOURCE	FM_ TAX_ RATES	TAX_ RATE_ DOMESTI C_ FEDERAL	nacionalfe deral	Y		
FM_ APPROX_ TAX_ BURDEN_ SOURCE	FM_ TAX_ RATES	TAX_ RATE_ IMPORTE D_ FEDERAL	importad osfederal	Y		
FM_ APPROX_ TAX_ BURDEN_ SOURCE	FM_ TAX_ RATES	TAX_ RATE_ STATE	estadual	Y		
FM_ APPROX_ TAX_ BURDEN_ SOURCE	FM_ TAX_ RATES	TAX_ RATE_ MUNICIP AL	municipal	Y		
FM_ APPROX_ TAX_ BURDEN_ SOURCE	FM_ TAX_ RATES	BEGIN_ DATE	vigenciain icio	Y		
FM_ APPROX_ TAX_ BURDEN_ SOURCE	FM_ TAX_ RATES	END_ DATE	vigenciafi m	Y		
FM_ APPROX_ TAX_ BURDEN_ SOURCE	FM_ TAX_ RATES	KEY_ CODE	chave	N		
FM_ APPROX_ TAX_ BURDEN_ SOURCE	FM_ TAX_ RATES	VERSION_ CODE	versao	N		

**Table 14-6 (Cont.) S9T\_TMPL\_COLS\_DEF**

<b>TEMPLATE_</b> <b>KEY</b>	<b>WKSHT_</b> <b>KEY</b>	<b>COLUMN_</b> <b>KEY</b>	<b>COLUMN</b> <b>_NAME</b>	<b>MANDATORY</b>	<b>DEFAULT</b>	<b>REQUIRED_</b> <b>VISUAL_IND</b>
FM_ APPROX_ TAX_ BURDEN_ SOURCE	FM_ TAX_ RATES	SOURCE_ CODE	fonte	N		
FM_ APPROX_ TAX_ BURDEN_ SOURCE	FM_ TAX_ RATES	ACTION	Action	N	NEW	



---

# Understanding RTIL WorkManager Settings

This appendix discusses the steps it is necessary to take to configure the RTIL WorkManager.

## Configuring RTIL WorkManager

The steps involved in configuring the RTIL WorkManager are as follows:

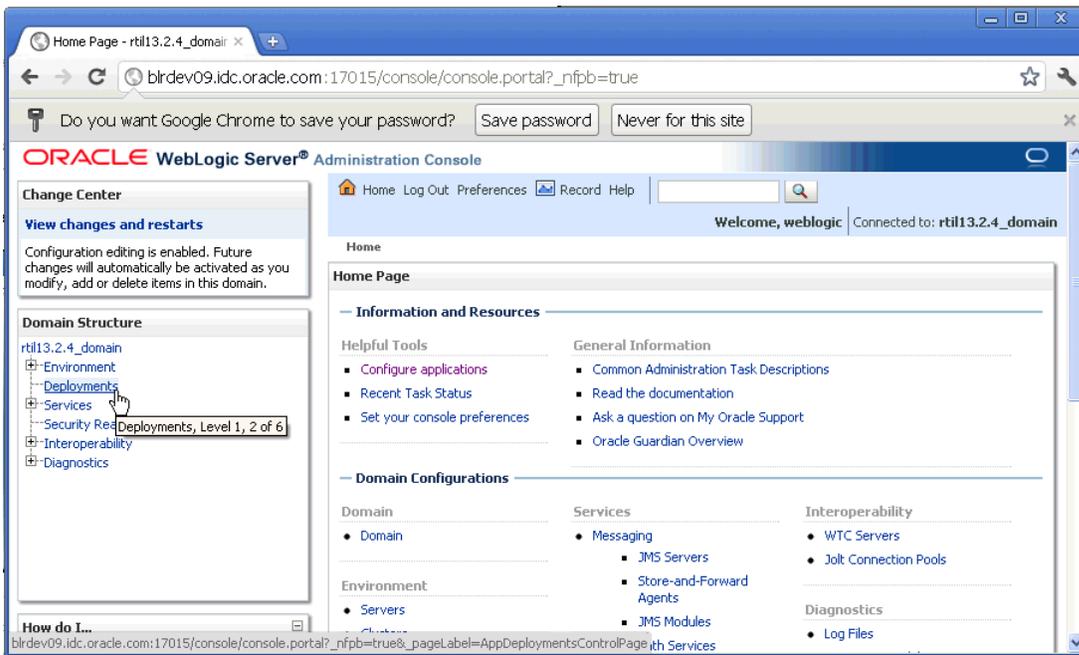
1. Login to Oracle WebLogic Server Administration Console.

*Figure A-1 Oracle WebLogic Server Login*



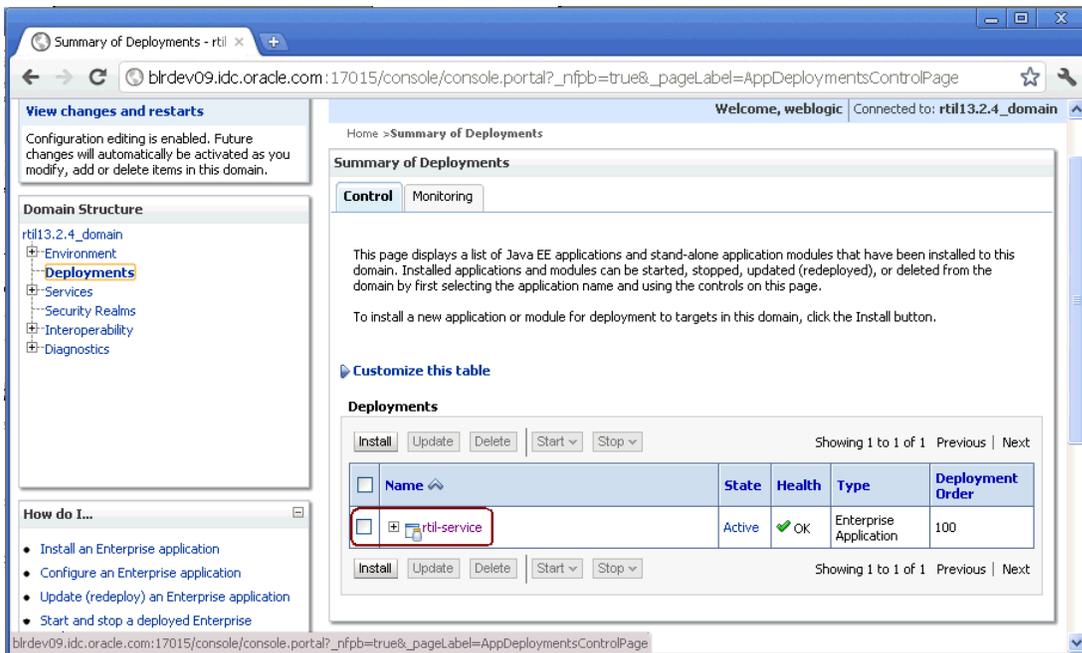
2. Click **Deployments** in the Domain Structure area of the console.

Figure A-2 Deployment



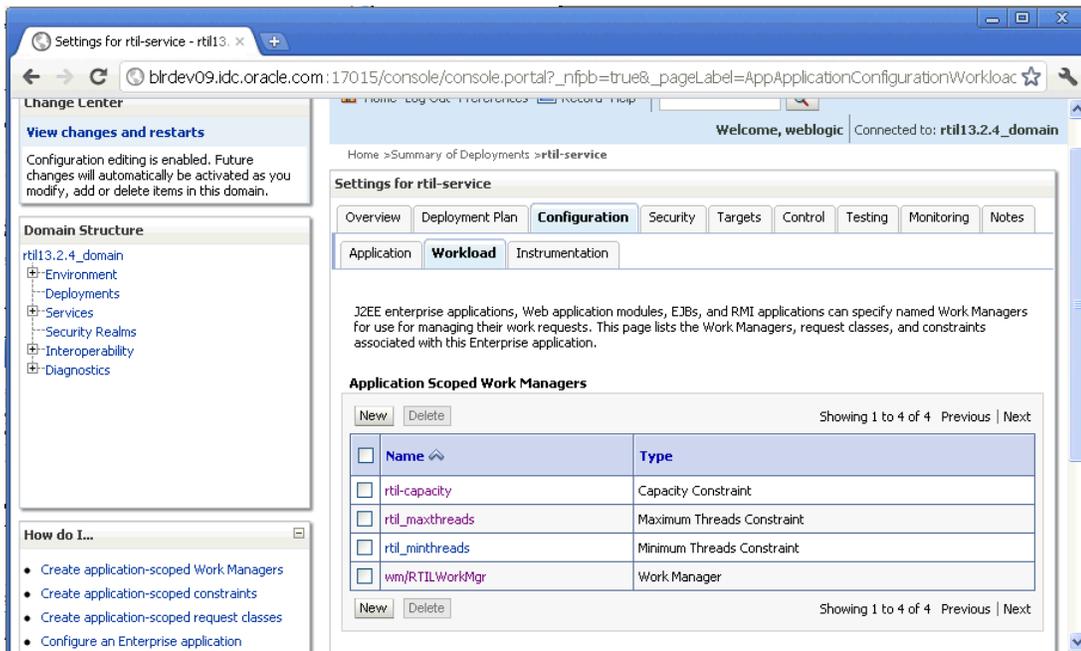
3. Select `rtil-service` from the Summary of Deployments area of the console.

Figure A-3 RTILService



4. In the Settings for `rtil-service` area of the console, from the **Configuration** tab, select **Workload**.

Figure A-4 Workload



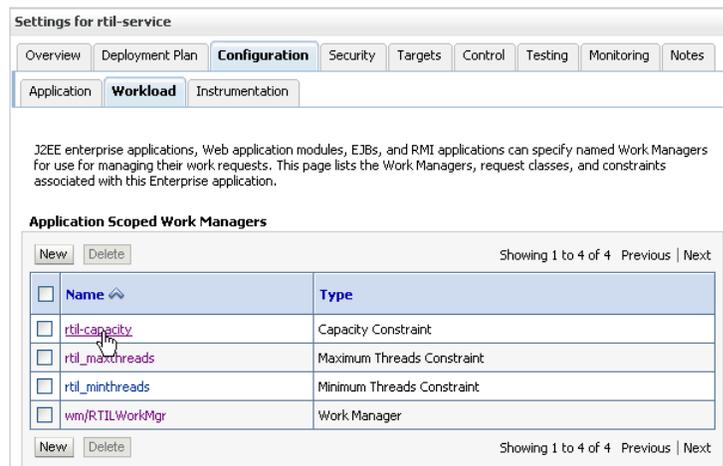
- The Applications Scoped Work Managers area includes `rtil_capacity`, `rtil_maxthreads`, and `rtil_minthreads`. Select the parameter that needs to be modified.

Figure A-5 Applications



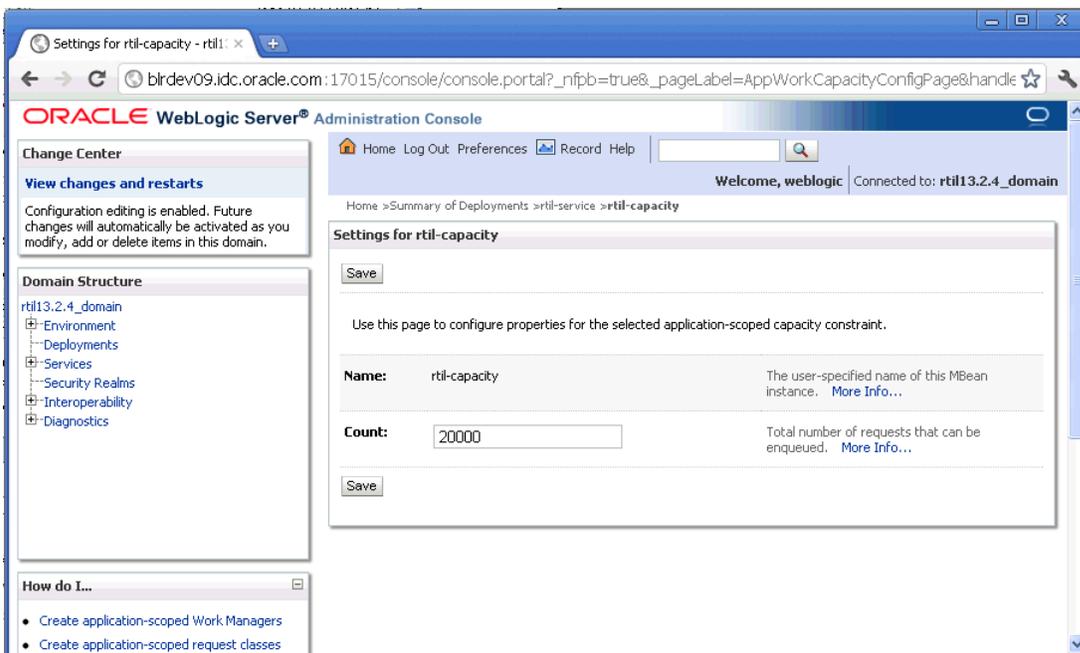
- If it is necessary to modify the number of requests that can be queued, click `rtil_capacity`.

**Figure A-6 RTIL Capacity**



7. In the Settings for rtil\_capacity window, enter the total number of requests that can be in the queue, and click **Save**.

**Figure A-7 Set RTIL Capacity**



8. It returns to the Applications Scoped Work Managers window. If necessary, modify the maximum number of concurrent threads that can execute requests, click **rtil\_maxthreads**.

**Figure A-8 RTIL Maxthreads**

Settings for rtil\_maxthreads

Save

Use this page to configure properties for the selected maximum threads constraint.

<b>Name:</b>	rtil_maxthreads	The user-specified name of this MBean instance. <a href="#">More Info...</a>
<b>Count:</b>	<input type="text" value="16"/>	Maximum number of concurrent threads that can execute requests sharing this constraint. <a href="#">More Info...</a>
<b>Connection Pool:</b>	<input type="text"/>	Name of the connection pool whose size is taken as the max constraint. <a href="#">More Info...</a>

Save

9. In the Settings for rtil\_maxthreads area of the window, enter the maximum number of concurrent threads that can execute requests, and click **Save**.



---



---

## ORFM and Brazil Configuration Checklist

This appendix presents a checklist of the most important configurations required to have ORFM and Brazil Localization setup done. It also contains some examples of basic data that can be used to start initial configuration. These basic sample data are optional and are simple suggestion to have an initial environment setup.

The setup and configuration presented here are also present in the *ORFM User Guide*, *Oracle Retail Merchandising System with Brazil Localization Installation Guide*, and as a part of *Implementation Guide* content as well.

The setup for RWMS and SIM is also present in *Implementation Guides* for these modules but were consolidated here as part of this checklist.

### Setting up ORFM

The following topics are covered in this section:

- [Setting up RMS \(Brazil-On\) System Options](#)
- [Setting up Set-Of-Books ID](#)
- [Setting up General Ledger](#)
- [System Setup](#)
- [Commerce Anywhere](#)
- [Setting up SIM System Options for ORFM](#)
- [Setting up RWMS System Options for ORFM](#)
- [Configuring Printer](#)

### Setting up RMS (Brazil-On) System Options

In case of RMS Brazil-On, ensure the following system options are setup as per below guidelines:

**Table B-1 RMS (Brazil-On) System Options**

Table Name	System Option
SYSTEM_CONFIG_OPTIONS	BASE_COUNTRY_ID = 'BR', CURRENCY_CODE = 'BRL' MULTI_CURRENCY_IND = 'Y'
LOCALIZATION_CONFIG_OPTIONS	DEFAULT_TAX_TYPE = 'GTAX'

**Table B-1 RMS (Brazil-On) System Options**

Table Name	System Option
FUNCTIONAL_CONFIG_OPTIONS	ORG_UNIT_IND = 'Y' SUPPLIER_SITES_IND = 'Y'
PRODUCT_CONFIG_OPTIONS	FINANCIAL_AP = 'O' or 'A' or NULL <b>Note 1:</b> 'O' for EBS Integration, 'A' for RFI Integration and 'Null' if no financial system integrated. <b>Note 2:</b> Enable (Value 'Y') below products only if they are integrated with RMS/RFM else setup as 'N'. RWMS_IND = 'Y', SIM_IND = 'Y', OMS_IND = 'Y' REIM_IND = 'N' ALLOCATION_IND = 'Y' SALES_AUDIT_IND = 'Y'
FINANCIAL_UNIT_OPTIONS	GL_ROLLUP = 'D' <b>Note:</b> EIT had tested this with GL_ROLLUP value as 'D'
COUNTRY_ATTRIB	COUNTRY_ID = 'BR'
INV_MOVE_UNIT_OPTIONS	SHIP_RCV_STORE=No SHIP_RCV_WH=No
COUNTRY_L10N_EXT	After ORFM install, Country_L10n_EXT must have a record

## Setting up Set-Of-Books ID

Insert set\_of\_books\_id record for Brazil currency (BRL) in FIF\_GL\_SETUP table.

### Example B-1 Sample set of records to be inserted

```
Insert into fif_gl_setup (SET_OF_BOOKS_ID, LAST_UPDATE_ID, SEQUENCE1_DESC, SEQUENCE2_
DESC, SEQUENCE3_DESC, SEQUENCE4_DESC, SEQUENCE5_DESC, SEQUENCE6_DESC, SEQUENCE7_
DESC, SEQUENCE8_DESC, SEQUENCE9_DESC, SEQUENCE10_DESC, CATEGORY_ID, DELIVER_TO_
LOCATION_ID, DESTINATION_ORGANIZATION_ID, PERIOD_NAME, SET_OF_BOOKS_DESC, CURRENCY_
CODE) values
```

```
(25000, 123456, 'DummySeq1', 'DummySeq2', 'DummySeq3', 'DummySeq4', 'DummySeq5', 'DummySe
q6', 'DummySeq7', 'DummySeq8', 'DummySeq9', 'DummySeq10', 3040, 50001, 3211, null, 'Set of
Books Id - USD', 'USD');
```

```
Insert into fif_gl_setup (SET_OF_BOOKS_ID, LAST_UPDATE_ID, SEQUENCE1_DESC, SEQUENCE2_
DESC, SEQUENCE3_DESC, SEQUENCE4_DESC, SEQUENCE5_DESC, SEQUENCE6_DESC, SEQUENCE7_
DESC, SEQUENCE8_DESC, SEQUENCE9_DESC, SEQUENCE10_DESC, CATEGORY_ID, DELIVER_TO_
LOCATION_ID, DESTINATION_ORGANIZATION_ID, PERIOD_NAME, SET_OF_BOOKS_DESC, CURRENCY_
CODE) values
```

```
(26000, 123457, 'DummySeq1', 'DummySeq2', 'DummySeq3', 'DummySeq4', 'DummySeq5', 'DummySe
q6', 'DummySeq7', 'DummySeq8', 'DummySeq9', 'DummySeq10', 3042, 50002, 1131, null, 'Set of
Books Id - Brazil', 'BRL');
```

## Setting up General Ledger

Following are the instructions to setup General Ledger (GL):

### Setting up GL Account

Insert record in FIF\_GL\_ACCT table for the above created set of book IDs.

#### Example B-2 Sample set of records to be inserted

```

Insert into fif_gl_acct (PRIMARY_ACCOUNT,SET_OF_BOOKS_
ID,ATTRIBUTE1,ATTRIBUTE2,ATTRIBUTE3,ATTRIBUTE4,ATTRIBUTE5,ATTRIBUTE6,ATTRIBUTE7,AT
TRIBUTE8,ATTRIBUTE9,ATTRIBUTE10,ATTRIBUTE11,ATTRIBUTE12,ATTRIBUTE13,ATTRIBUTE14,AT
TRIBUTE15,DESCRIPTION1,DESCRIPTION2,DESCRIPTION3,DESCRIPTION4,DESCRIPTION5,DESCRIP
TION6,DESCRIPTION7,DESCRIPTION8,DESCRIPTION9,DESCRIPTION10,DESCRIPTION11,DESCRIPTI
ON12,DESCRIPTION13,DESCRIPTION14,DESCRIPTION15) values
(1234567,111,'1','1','1','NETSALEDR',null,null,null,null,null,null,null,null,
null,null,'Company','Area','Dept','Account',null,null,null,null,null,null,null,nu
ll,null,null,null);
Insert into fif_gl_acct (PRIMARY_ACCOUNT,SET_OF_BOOKS_
ID,ATTRIBUTE1,ATTRIBUTE2,ATTRIBUTE3,ATTRIBUTE4,ATTRIBUTE5,ATTRIBUTE6,ATTRIBUTE7,AT
TRIBUTE8,ATTRIBUTE9,ATTRIBUTE10,ATTRIBUTE11,ATTRIBUTE12,ATTRIBUTE13,ATTRIBUTE14,AT
TRIBUTE15,DESCRIPTION1,DESCRIPTION2,DESCRIPTION3,DESCRIPTION4,DESCRIPTION5,DESCRIP
TION6,DESCRIPTION7,DESCRIPTION8,DESCRIPTION9,DESCRIPTION10,DESCRIPTION11,DESCRIPTI
ON12,DESCRIPTION13,DESCRIPTION14,DESCRIPTION15) values
(1,111,'1110','1111','1112','1113','1114','D','1','1','1','1',null,null,null,null,
null,null,null,null,null,null,null,null,null,null,null,null,null,null,null);
    
```

### Setting up Chart of Account

Following are the instructions to setup Chart of Account.

#### Example B-3 Sample set of records to be inserted

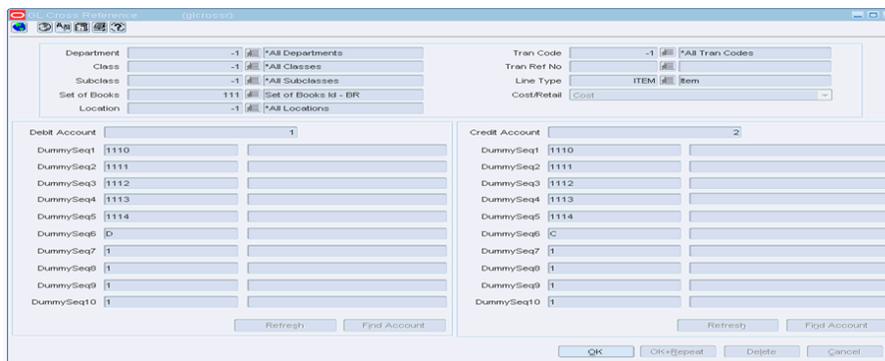
```

Insert INTO FM_COA_SETUP VALUES (1,1,'Natural
Account','Company','Department','Class','Subclass','Location','Transfer
Entity',NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,
NULL,NULL,NULL,NULL,NULL,NULL,NULL);
    
```

### Setting up GL Cross-Reference

Setup GL Cross Reference for all transaction codes and merchandise hierarchy. You need to specify what transaction code falls under which account type. In RMS Navigate to **Finance**, then **GL Cross Reference**, and then **New**.

Figure B-1 Sample setup in UI for all Tran codes for all Items and all Departments



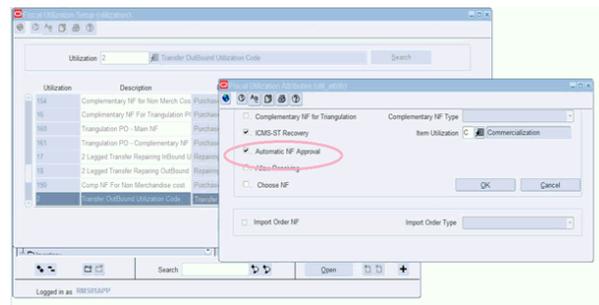
## System Setup

ORFM system setup involves the following: System Options, Fiscal Number, Tolerances, and NF-e Configuration setup.

### System Options

- Initial setup is provided through Control scripts that were run during installation. This initial setup allows RFM, after installation process, perform all functions appropriately. System Options that refers fiscal utilizations and document type, all combinations are already in system.
- System administration is permitted modify any ORFM system options values to match business needs, but make sure to avoid invalid mask/invalid format error.
- Create a Partner in RMS and set the values in the following columns to avoid invalid partner id error while receiving the PO in RWMS:
  - FM\_SYSTEM\_OPTIONS.EDI\_DEF\_PTNR\_TYPE= 'E'
  - FM\_SYSTEM\_OPTIONS.EDI\_DEF\_PTNR\_ID= partner\_id
- For the transactions like RTV, Transfer, Inv Adj, and XC Orders whether the NF generated must be in Validated/ Approved status depends on the attribute AUTO\_APPROVE\_IND set for the particular transaction's util\_id.
- Setup 2 transfer entities (org units) in RMS - 1 for intercompany and 2nd for intracompany transfers.

**Figure B–2 Fiscal Utilization Setup Window**



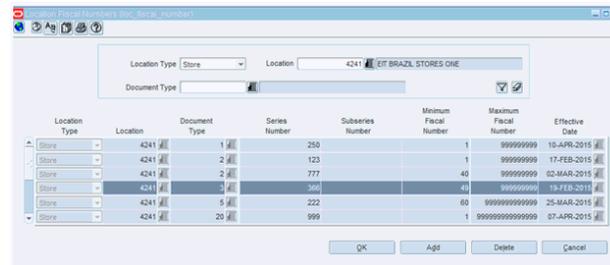
**Figure B–3 Corresponding Record in FM\_UTILIZATION\_ATTRIBUTES**

```
select * from FM_UTILIZATION_ATTRIBUTES where utilization_id=2;
```

UTILIZATION_ID	COMP_NF_IND	ALLOW_RECEIVING_IND	ICMSST_RECOVERY_IND	CHOOSE_DOC_IND	AUTO_APPROVE_IND	CREATE_DATETIME	CREATE_ID	LAST_UPDATE_DATETIME
2						10-FEB-15	RMS01APP	06-MAR-15

### Fiscal Number Setup:

After locations are created in RMS and ORFM, ensure that all the doc types are ranged to each store and warehouse location. You must also provide minimum and maximum fiscal number values.

**Figure B-4 Location Fiscal Numbers Window**

## Tolerances

Ensure that for each supplier site, tolerance values are setup for both cost and quantity.

**Figure B-5 Tolerances Window**

## Commerce Anywhere

Ensure that customer order created from external order management system should be a numeric avoiding alpha numeric.

## Setting up SIM System Options for ORFM

For the SIM to send the pre-shipment message for various transactions, following fields in CONFIG\_STORE table should be set to true:

- PRESHIPMENT\_NOTIFICATION=true
- PRESHIPMENT\_NOTIFICATION\_RETURN=true
- PRESHIPMENT\_NOTIFICATION\_TRANSFER=true

For Over-receiving/Under receiving in SIM, whether to accept the quantity more/less than the ordered quantity depends on the following options which has to be set in CONFIG\_STORE table:

- CONFIG\_STORE.DIRECT\_DELIVERY\_REMOVE\_OVER\_RECEIVE='TRUE'(removes the excess qty than the ordered Qty)
- CONFIG\_STORE.DIRECT\_DELIVERY\_REMOVE\_DAMAGES='TRUE'(removes the excess damaged qty than the ordered Qty)

---

**Note:** A Return NF will be created in ORFM for the excess Qty than the ordered Qty, incase if the above parameters are set to false.

---

## Setting up RWMS System Options for ORFM

1. To enable Brazil localization in RWMS, the following system options should be set up for all facilities:

Set LOCAL\_BRAZIL and LOC\_BRAZIL\_AUTO\_RECV to Y in SCP table

2. While creating Appointment for PO in RWMS, Schedule Number field should be enabled in ORFM environment. Below is the field in SCP table which should be set to Y in order to enable that field for all facilities.

```
update scp set scp_current_val='Y' where facility_id='facility id' and scp_name='recv_schedule_nbr';
```

## Configuring Printer

It is necessary to set up Warehouse and Store Printers in BI publisher for printing and let NF status to Warehouse and Store personal so that shipping/receiving of inventory can be completed at respective locations.

The steps to configure warehouse and store printer (network printers) in BI publisher: are as follows:

### RMS/ORFM

1. Login to **BI Publisher** as administrator.
2. Navigate to **Printer Configuration**.
3. Click **Add Printer** and printer URI name and filter.

---



---

**Note:** Network administrator should be able to provide printer URI information.

---



---

4. Click **Save**.

The steps to print NF status to warehouse or store printer setup in BI publisher are as follows:

5. Login to RMS/ORFM, navigate to **Fiscal Management > Receiving/Issue**.
6. Select a schedule in approved status and open Nota Fiscal.
7. On **Fiscal Doc** form, click **Options > Print Document**, BI publisher form appears with fiscal document details.
8. Go to **Output**, click **Add Destination**, select **Printer Group**, and **Printer** (configured with above steps) then click **Submit**.

---



---

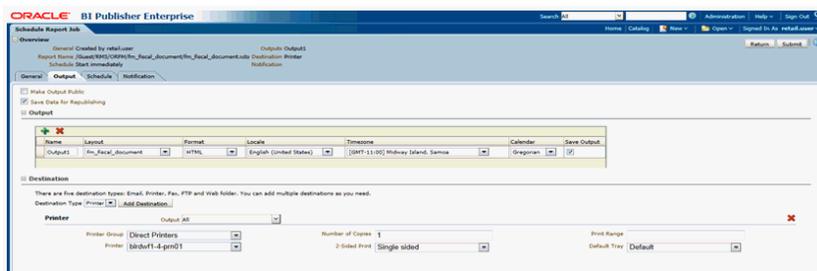
**Note:** The printer configured here will be setup in RWMS/SIM (details below) as well for it to send printer info to ORFM (part of ship info RIB message) so that ORFM can print NF status to same printer to let RWMS personnel know of NF status and to continue with rest of the shipping or receiving process.

---



---

**Figure B-6 BI Publisher Enterprise Window**



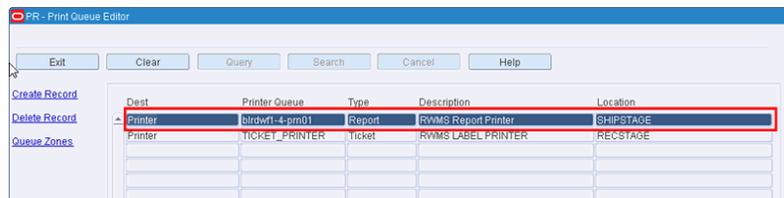
**RWMS**

The steps to configure printer (setup in BI publisher) in RWMS are as follows:

**Note:** Part of customer order packing (for wh-c customer order shipping), this printer should be used in packing screen (Processing > Packing).

1. Login to RWMS, navigate to **Setup - Administration > Print Queue Editor**.
2. Click **Create Record** > Fill detail Dest=printer, Type=Report, Printer Queue=description=printer name (setup in BI publisher), location=SHIPSTAGE

**Figure B-7 TR - Print Queue Editor Window**



**SIM**

The steps to configure printer (setup in BI publisher) in SIM are as follows:

1. Login to SIM, navigate to **Admin -> Print Setup-> Printers**.
2. Click **Add** > Add a Description, Type='PostScript', Network Address= IP Address of printer setup in BI publisher and click **Save**.

**Figure B-8 Store Inventory Management Window**



